

VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA  
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

DIPLOMOVÁ PRÁCE

VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA  
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

KATEDRA ELEKTRONIKY

Laboratorní stanoviště pro předmět Signálové procesory

Laboratory Station for the Subject Signal Processors

Student: Bc. Jan Šmajstrla

Vedoucí diplomové práce: doc. Ing. Petr Palacký, Ph.D.

Ostrava 2011

VŠB - Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra elektroniky

## Zadání diplomové práce

Student: **Bc. Jan Šmajstrla**  
Studijní program: N2649 Elektrotechnika  
Studijní obor: 2612T015 Elektronika  
Téma: **Laboratorní stanoviště pro předmět Signálové procesory**  
**Laboratory Station for the Subject Signal Processors**

Zásady pro vypracování:

1. Proveďte popis vlastností signálových procesorů z hlediska využití pro číslicové zpracování signálů.
2. Navrhněte a realizujte laboratorní stanoviště pro návrhy a implementace číslicového zpracování signálů pomocí signálového procesoru.
3. Proveďte měření na realizovaných laboratorních úlohách a vypracujte průvodní zprávu s návody pro výuku.

Seznam doporučené odborné literatury:

Dle pokynů vedoucího závěrečné práce.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Ing. Petr Palacký, Ph.D.**

Datum zadání: 19.11.2010

Datum odevzdání: 06.05.2011



doc. Ing. Petr Palacký, Ph.D.  
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

„Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně a uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.“

Datum: 3. 5. 2011

podpis:

## **Abstrakt**

Tato diplomová práce má sloužit jako učební pomůcka pro předmět Signálové procesory, proto se zabývá rozbořením použití a implementací algoritmu. Popisuje vnitřní strukturu procesoru TMS320F2812, jeho nastavení a také využití. Největší díl je však věnován číslicovým filtrům. Diplomová práce tak čtenáře seznamuje s jednotlivými typy používaných filtrů jak z teoretického, tak z praktického hlediska. Uvádí také jednotlivé implementace číslicových filtrů na procesor TMS320F2812. Pro názornost jsou v závěru uvedeny příklady na výpočet daných filtrů. Přílohy tvoří návody na cvičení včetně vzorových protokolů.

## **Klíčová slova**

Signálový procesor, filtr, Event Manager, IIR filtr, FIR filtr, Butterwortová aproximace, Čebyševová aproximace, Eliptické aproximace, Fourierovy řady

## **Abstracts**

This thesis is intended as an educational tool for the subject signal processors, therefore, deals with the analysis of the use and implementation of the algorithm. It describes the internal structure of the TMS320F2812 processor, setup and use. The largest part is devoted to digital filters. This thesis introduces the reader to the various types of filters used in both theoretical and practical perspective. It also lists the various implementations of digital filters on the TMS320F2812 processor. To illustrate examples of the end for the calculation of the filter. Attachments are instructions for the exercises, including standard protocols.

## **Keywords**

Signal processing, filter, Event Manager, IIR filter, FIR filter, Butterworth approximation, Chebyshev approximations, Elliptic approximations, Fourier series

# SEZNAM POUŽITÝCH ZKRATEK

A/D	analogově digitální převodník
ALU	aritmetickologická jednotka
ARAU	jednotka generování adres
CPU	centrální řídicí jednotka
D/A	digitálně analogový převodník
DSP	digitální signálový procesor
EV	Event Manager
EVA	Event Manager A
EVB	Event Manager B
FIFO	vyrovnávací paměť
FIR	filtr s konečnou impulzní odezvou
Flash	paměť typu ROM, lze mazat a programovat
GPIO	označení vstupně/výstupního portu určeného pro všeobecné použití
IIR	filtr s nekonečnou impulzní odezvou
JTAG	rozhraní na testování signálu na vývodech integrovaného obvodu
MAC	násobička a sčítačka
OZ	operační zesilovač
p	operátor Laplaceovy transformace
PIE	jednotka periferního přerušení
PWM	pulsně šířková modulace
S/H	Sample/Hold
SIMD	paralelní zpracování dat
SPI	rozhraní pro sériovou komunikaci s periferními obvody
vf	vysokofrekvenční signál
z	operátor Z-transformace

A	zesílení	[-]
f	frekvence	[Hz]
$f_v$	vzorkovací frekvence	[Hz]
$H_a(p)$	přenosová funkce analogového přenosu	[-]
$h[n]$	impulsní charakteristika diskrétního signálu	[-]
$H(z)$	přenosová funkce diskrétního přenosu	[-]
$H(e^{j\omega})$	komplexní kmitočtová charakteristika	[-]
T	perioda	[s]
$T_v$	vzorkovací perioda	[s]

$U$	napětí	[V]
$\omega$	kruhová frekvence	[rad <sup>-1</sup> ]
$\varepsilon$	parametr zvlnění v propustném pásmu	[-]

# OBSAH

<b>ÚVOD .....</b>	<b>1</b>
<b>1     <b>SIGNÁLOVÝ PROCESOR.....</b></b>	<b>2</b>
<b>1.1 Architektura procesorového jádra C28x .....</b>	<b>3</b>
1.1.1     Paralelní zpracování dat.....	3
1.1.2     Zřetěžené zpracování instrukcí .....	4
<b>1.2 Časovače Timer 0/1/2.....</b>	<b>4</b>
<b>1.3 Přerušovací systém.....</b>	<b>5</b>
<b>1.4 Event Manager .....</b>	<b>7</b>
<b>1.5 Generování signálů PWM pomocí TMS320F2812.....</b>	<b>9</b>
1.5.1     Popis tvorby PWM modulace pomocí EV jednotky .....	9
<b>1.6 Práce s A/D a D/A převodníkem .....</b>	<b>10</b>
1.6.1     A/D převodník .....	10
1.6.2     Sériové periferní rozhraní (SPI).....	11
1.6.3     D/A převodník DAC 7715 .....	12
<b>1.7 Vstupně-výstupní porty GPIO .....</b>	<b>13</b>
<b>2     <b>ROZŠÍŘUJÍCÍ DESKA.....</b></b>	<b>14</b>
<b>2.1 A/D vstup .....</b>	<b>14</b>
<b>2.2 D/A převodník .....</b>	<b>14</b>
<b>2.3 Napájení .....</b>	<b>15</b>
<b>2.4 Konektory .....</b>	<b>17</b>
<b>3     <b>FILTRY .....</b></b>	<b>19</b>
<b>3.1 Kauzalita, linearita a stabilita .....</b>	<b>19</b>
<b>3.2 Z-transformace.....</b>	<b>19</b>
<b>3.3 Diferenční rovnice .....</b>	<b>20</b>
<b>3.4 Filtry FIR.....</b>	<b>21</b>
3.4.1     Podmínka lineární fáze filtru .....	22
3.4.2     Postup návrhu FIR filtru pomocí Fourierových řad s využitím funkce oken.....	23
3.4.3     Okna používaná pro návrh filtru FIR.....	25
3.4.3.1 <i>Obdélníkové okno.....</i>	26
3.4.3.2 <i>Bartletovo okno .....</i>	26
3.4.3.3 <i>Hannovo okno .....</i>	27
3.4.3.4 <i>Hammingovo okno .....</i>	27
3.4.3.5 <i>Blackmanovo okno .....</i>	28
3.4.3.6 <i>Srovnání oken.....</i>	28
<b>3.5 Filtry IIR.....</b>	<b>29</b>
3.5.1     Metody návrhu číslicových filtrů IIR.....	29
3.5.2     Bilineární transformace.....	30



3.5.3	Butterworthova aproximace .....	32
3.5.4	Čebyševová aproximace .....	33
3.5.5	Inverzní Čebyševová aproximace .....	33
3.5.6	Elíptická aproximace .....	33
3.5.7	Postup návrhu IIR filtru .....	33
<b>4</b>	<b>ALGORITMY A ZPRACOVÁNÍ JEDNOTLIVÝCH FILTRŮ .....</b>	<b>34</b>
4.1	Vyjádření čísla v pevné řádové čárce .....	34
4.2	Implementace FIR filtru .....	35
4.2.1	Implementace FIR filtru v jazyce Assembler .....	36
4.3	Výpočet koeficientu pro FIR filtr .....	39
4.3.1	Program pro výpočet koeficientu pro FIR filtr .....	39
4.3.1.1	Popis hlavního panelu programu .....	40
4.3.1.2	Popis programové struktury vytvořené v LabView .....	42
4.4	Implementace IIR filtru .....	45
<b>5</b>	<b>MĚŘENÍ .....</b>	<b>50</b>
5.1	Příklad výpočtu koeficientů FIR filtru .....	50
5.2	Změřené průběhy filtru FIR .....	54
5.3	Příklad výpočtu koeficientů IIR filtru .....	56
5.4	Změřené průběhy filtru IIR .....	59
<b>6</b>	<b>ZÁVĚR .....</b>	<b>60</b>
<b>7</b>	<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>61</b>
<b>8</b>	<b>SEZNAM PŘÍLOH .....</b>	<b>62</b>

# ÚVOD

Z důvodu neustálého zpracovávání digitálních signálů, jako je video, audio atd., bylo nutné vyvinout procesor, který by byl schopen rychle zpracovat velký objem dat. Proto se začala vyrábět vlastní větev mikroprocesorů. Ta se nejprve vyráběla jen jako holé výpočetní jádro a teprve až časový vývoj přinesl nutnost tyto procesory vyrábět s perifériemi přímo na jednom čipu. Dnes jsou signálové procesory využívány v nejrůznějších aplikacích. Zejména při použití vektorové regulace, kde je třeba pracovat v reálném čase. Pro jejich velký výpočetní výkon se tak uplatnily i pro aplikace výkonových měničů. Tato diplomová práce se zabývá právě signálovými procesory, především se pak zaměřuje na procesor TMS320F2812 od firmy Texas Instrument.

Jelikož je k výuce předmětu Signálové procesory používán vývojový kit právě s procesorem TMS320F2812, klade si tato diplomová práce za cíl být užitečnou učební pomůckou uvítanou především mezi studenty již zmíněného předmětu. Čtenáři se však mohou dočíst nejen o signálovém procesoru, ale také o rozšiřující desce či o číslicových filtrech, neboť jsou používány v laboratorním stanovišti.

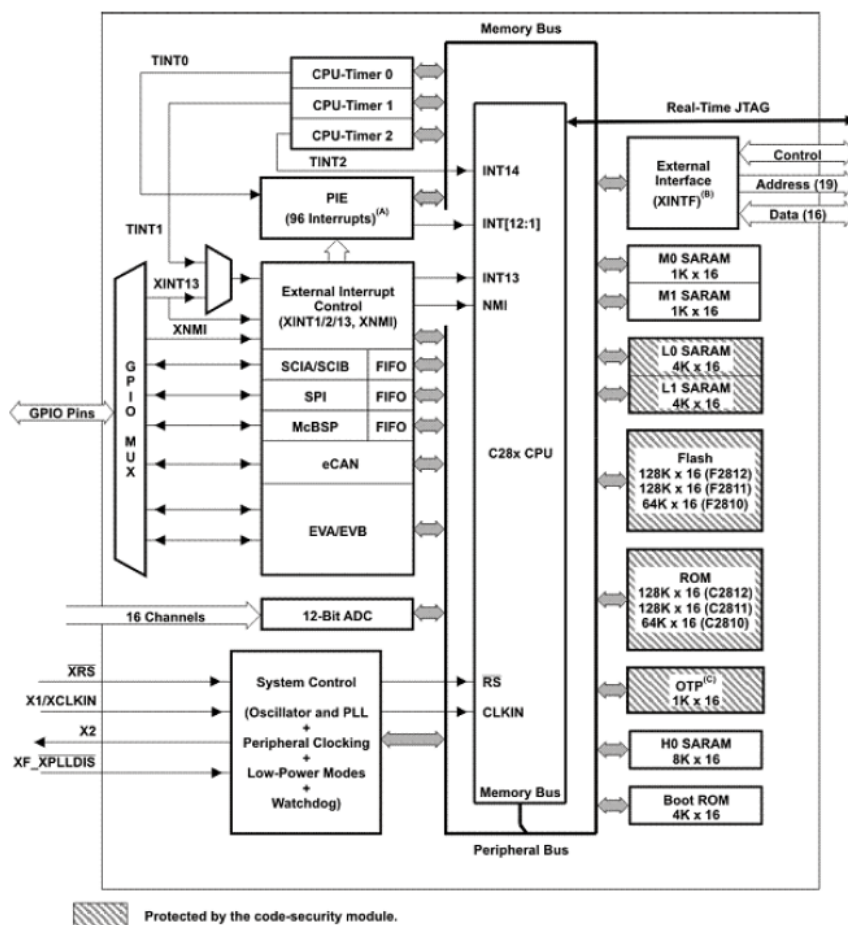
Diplomová práce má 5 kapitol. V první kapitole je definován a popsán samotný procesor. Je zde uvedena a pomocí blokových schémat i zobrazena jeho základní struktura a popsány nastavovací parametry, kterými jsou například časovače Timer 0/1/2, SPI, boot ROM, A/D převodník či paměti typu flash. Druhá kapitola popisuje rozšiřující desku pro vývojový kit, jež má samostatné napájení a rozšiřuje tak možnosti některých periférií. Je zde vysvětlen postup úpravy A/D převodníku nacházejícího se na procesoru TMS320F2812. Jelikož samotný procesor a ani vývojový kit neobsahuje D/A převodník, bylo nutné tento převodník na rozšiřující desku umístit. Mimo D/A převodníku je zde také zapotřebí umístit napájecí větev pro napěťový posun invertujících zesilovačů. Proto v kapitole nemůže chybět ani popis D/A převodníku a napájení. Vše je doprovázeno přehlednými vysvětlujícími schématy týkající se především zapojení uvedených periférií. Třetí kapitola je zaměřena na číslicové filtry, tj. na FIR filtry a IIR filtry. Pozornost je zde věnována především jejich popisu, základní struktuře a teoretickému odvození, dále tzv. oknům, která se používají pro návrh filtru FIR a metodám návrhu IIR filtru. Postup při implementaci jednotlivých filtrů na procesor TMS 320F2812 je znázorněn ve čtvrté části této diplomové práce. Je zde představen i program pro výpočet koeficientů FIR. Poslední kapitola ukazuje, jak lze jednotlivé filtry spočítat. Přílohy pak tvoří návody na cvičení včetně vzorových protokolů.

# 1 SIGNÁLOVÝ PROCESOR

Signálový procesor je mikroprocesor uzpůsobený pro efektivní realizaci algoritmů číslicového zpracování signálů v reálném čase. To vše s nejmenšími vynaloženými náklady.

Signálové procesory mají modifikovanou Harvardskou strukturu umožňující přenos hodnot z programové paměti do datové a naopak. Pro dosažení vysoké rychlosti výpočtu se užívá zvýšení hodinového kmitočtu a paralelní zpracování dat s obvodovou podporou cyklů opakování. Obsahují více sběrnic než klasické mikroprocesory, aby bylo možné provádět větší množství operací najednou.

Tato diplomová práce se zaměřuje na signálový procesor TMS320F2812 od firmy Texas Instrument, který je určený především k použití pro výkonové měniče. Jeho výbava tomu také odpovídá. Základem procesoru je výpočetní jádro označované jako C28x v sobě skrývající aritmeticko-logickou jednotku (ALU), paralelní posuvnou jednotku (Shifter), násobičku a sčítačku (MAC) a generátor adres (ARAU). Na toto jádro jsou napojeny další periférie, tedy paměti typu flash, ROM, SARAM L0/L1/M0/M1/H0, boot ROM, časovače Timer 0/1/2, jednotky PIE, EVA, EVB, A/D převodník, JTAG test/emulátor, SPI a další. Celkové blokové schéma procesoru TMS320F2812 je vyobrazeno na obrázku 1.1.



Obr. 1.1: Bloková struktura TMS320F2812 [9]

## 1.1 Architektura procesorového jádra C28x

Ve většině operací umožňuje jádro C28x vykonat potřebnou operaci v jednostrujovém cyklu.

### Sběrnice

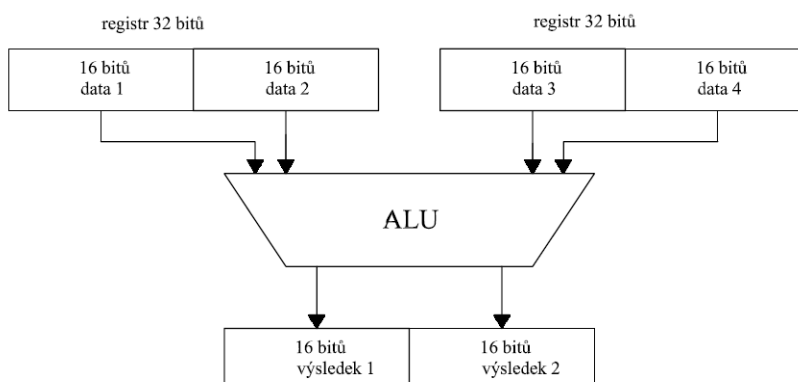
Procesor obsahuje šest sběrnic, z nichž jedna je 22-bitová a ostatní 32-bitové. Tyto sběrnice jsou rozděleny pro programovou a datovou paměť. Programová paměť má k dispozici dvě sběrnice. Jednu k zadání adresy, která je jediná 22-bitová a jednu datovou sběrnici. Datová paměť má k dispozici jednu datovou sběrnici a dvě adresové sběrnice, přičemž z jedné se čtou adresy a do druhé jsou adresy zapisovány. Poslední sběrnice je pro adresovou i datovou paměť společná. Jde o sběrnici pro zápis dat do těchto pamětí.

### Výpočetní jednotky

Aritmetická logická jednotka (ALU) provádí téměř všechny výpočty v jednostrujovém cyklu. Je 32-bitová a vykonává veškeré aritmetické operace, logické operace a operace posuvu. Pro rychlé zpracování násobení je v jádře obsažená násobící jednotka (Multiplier), která dokáže násobit dvě 32-bitové čísla v jednostrujovém cyklu, přičemž výsledné číslo může být až 64-bitové. Jelikož je uspořádání paměti 32-bitové, je nutné tak velké číslo rozdělit na dvě poloviny a uložit je jako 32-bitové. Posuvná jednotka (Shifter) umožňuje posouvání operandu nezávisle na ostatních jednotkách a tak může operand upravit již před jeho posláním k dalšímu zpracování. Jednotka ARAU pak zajišťuje řízení adresových registrů.

#### 1.1.1 Paralelní zpracování dat

Ke zvýšení výpočetního výkonu se s oblibou používá koncepce označovaná jako SIMD (Single-Instruction, Multiple-Data). Tato koncepce umožňuje zpracovat větší množství stejných výpočetních operací pro více navzájem nezávislých dat, a to během jednoho instrukčního cyklu. Proto procesor může s registry pracovat tak, že například 32-bitový registr se rozdělí na dva nezávislé 16-ti bitové registry.

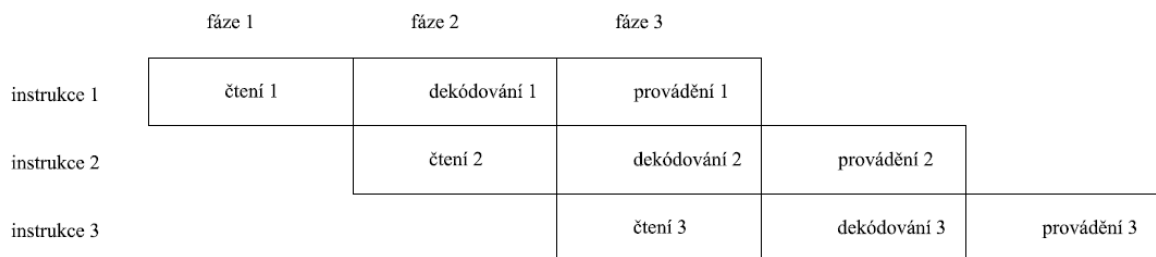


*Obr. 1.2: Princip paralelního zpracování*

S výše znázorněným zpracováním dat se lze setkat při provádění algoritmu číslcových filtrů. U filtrů se provádí několik po sobě stejných operací a tudíž je výhodné použít tuto metodu pro zkrácení výpočtových operací.

### 1.1.2 Zřetěžené zpracování instrukcí

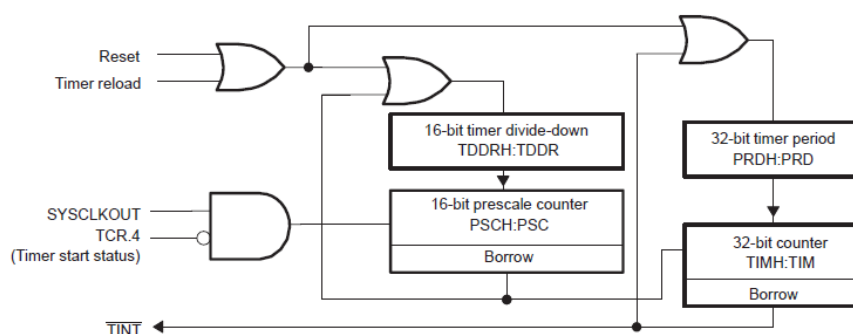
O zřetěženém zpracování instrukcí se nejčastěji mluví jako o Pipelingu. Vychází z faktu, že při zpracování jedné operace je nutné použít několik po sobě funkčních jednotek, které jsou rychlé, účelné, ale určené jen pro určitou část operace. Tedy, jednotky budou pracovat pouze určitou část operace a pak zůstanou nevyužity. Důsledkem tohoto je nevyužití plného výpočetního výkonu. Jelikož je však důležité co nejvíce výpočetní výkon využít, jsou procesory konstruovány tak, aby jednotky byly stále využity. Programová instrukce je proto rozdělena na několik částí, přičemž každá část je vykonávána v jiné jednotce. To znamená, že první instrukce je vykonávána v první jednotce, pak se přesouvá do druhé jednotky a po přesunu do druhé jednotky je už druhá instrukce nahrávána do první jednotky.



**Obr. 1.3: Zřetěžené zpracování dat**

## 1.2 Časovače Timer 0/1/2

Na mikroprocesoru se nacházejí tři 32-bitové univerzální časovače. Timer 0, který je připojený na jádro přes jednotku přerušení PIE. Dále Timer 1, jenž je přímo spojený s jádrem a Timer 2, který je připojený k jádru přes blok MUX. Časovače Timer 1 a Timer 2 jsou rezervovány pro BIOS. Z toho důvodu se nedají použít pro uživatelské aplikace. Pouze Timer 0 může být použit při programování algoritmů. Obrázek 1.4 znázorňuje blokovou strukturu všech tří časovačů, které bývají stejné i pro jiné mikroprocesory firmy Texas instrument.



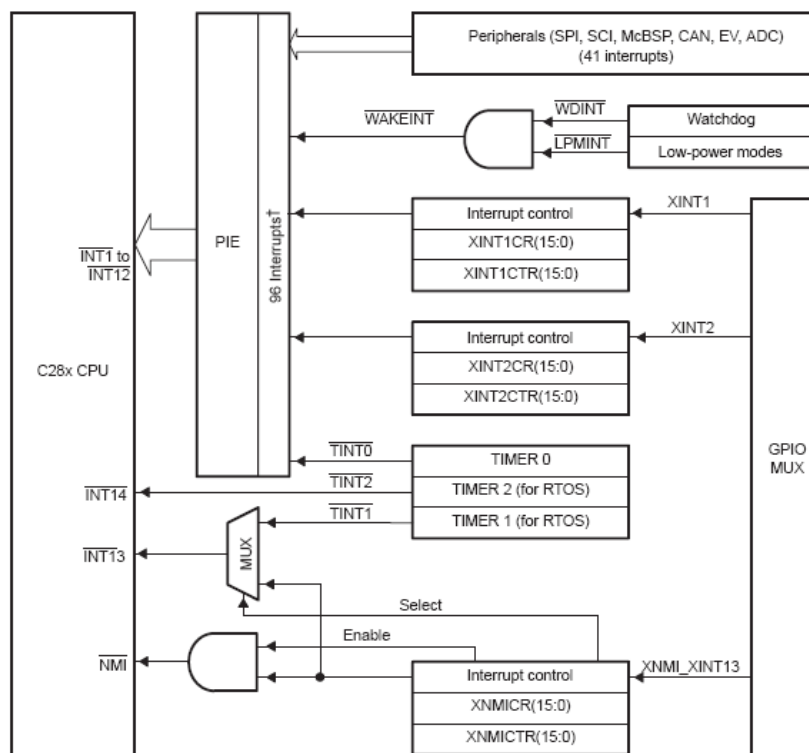
**Obr.1.4: Bloková struktura časovačů [9]**

Na výše uvedeném obrázku lze vidět, že systémový hodinový signál, v obrázku označený jako SYCLKOUT, přichází nejprve do děličky s 16-ti bitovým čítačem s nastavovacím bitem PSC. Tento 16-ti bitový čítač obsahuje nastavovací registr TDDR. Při průchodu nulou je vygenerován hodinový signál pro 32-bitový čítač, který se nastavuje registrem PRD. Z tohoto čítače je pak generován signál pro přerušení TINT. Časovače znázorněné na obrázku jen inkrementují a při dosažení výsledné hodnoty skočí opět na nulu.

### 1.3 Přerušovací systém

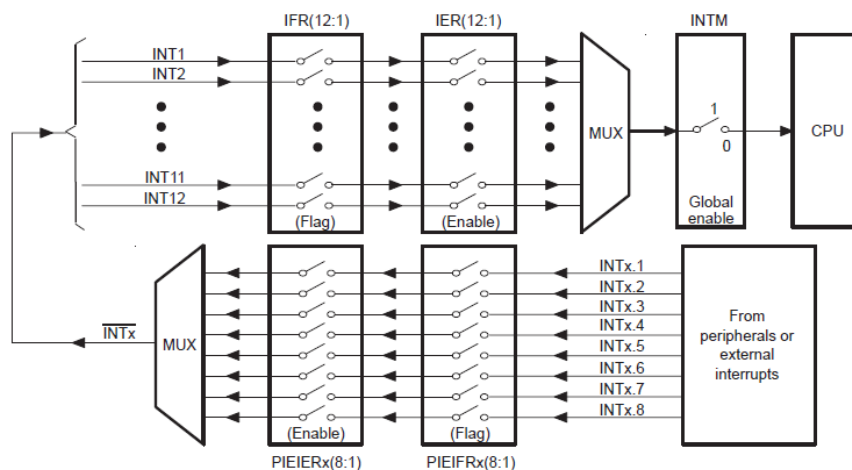
Signálový procesor TMS320F2812 má určitý počet zdrojů přerušení, které mohou být povolená nebo zakázána pomocí registru PIE Vector Table. Níže uvedené blokové schéma (obrázek 1.5) znázorňuje strukturu zdrojů přerušení.

Periferní přerušovací blok (PIE) podporuje 96 individuálních přerušení seskupených do 12-ti bloků po osmi přerušeních. Těchto 12 bloků se posílá do CPU jako 12 přerušení pod označením INT1 až INT12. Další přerušení posílaná do CPU jsou od časovače Timer 1 (INT13) a Timer 2 (INT14).



**Obr.1.5: Zdroje přerušení [9]**

K tomu, aby mikroprocesor mohl provést požadované přerušení, je třeba povolit celou cestu od periférie až k CPU. Celá cesta přerušení je naznačena na obrázku 1.6 , na němž je také uvedeno, které registry se musí nastavit, aby přerušení bylo provedeno.



**Obr. 1.6: Přerušovací cesta [9]**

K nastavení přerušování se vychází z tabulky 1.1 Tabulka znázorňuje, na jakém vstupu je dané zařízení a jaké cesty se musí povolit, aby bylo možné přerušování vykonat.

Pro snadnější pochopení, jak nastavit např. přerušování od časovače Timer 0, je uveden následující postup. Nejprve je třeba zjistit, kde se časovač Timer 0 nachází. Z tabulky lze vyčíst, že je to na řádku označeném INT 1 a ve sloupci INTx.7. INT 1 a INTx.7 představují bity, jejichž povolením se nastaví cesta přerušování. Nakonec se povolí globální přerušování INTM.

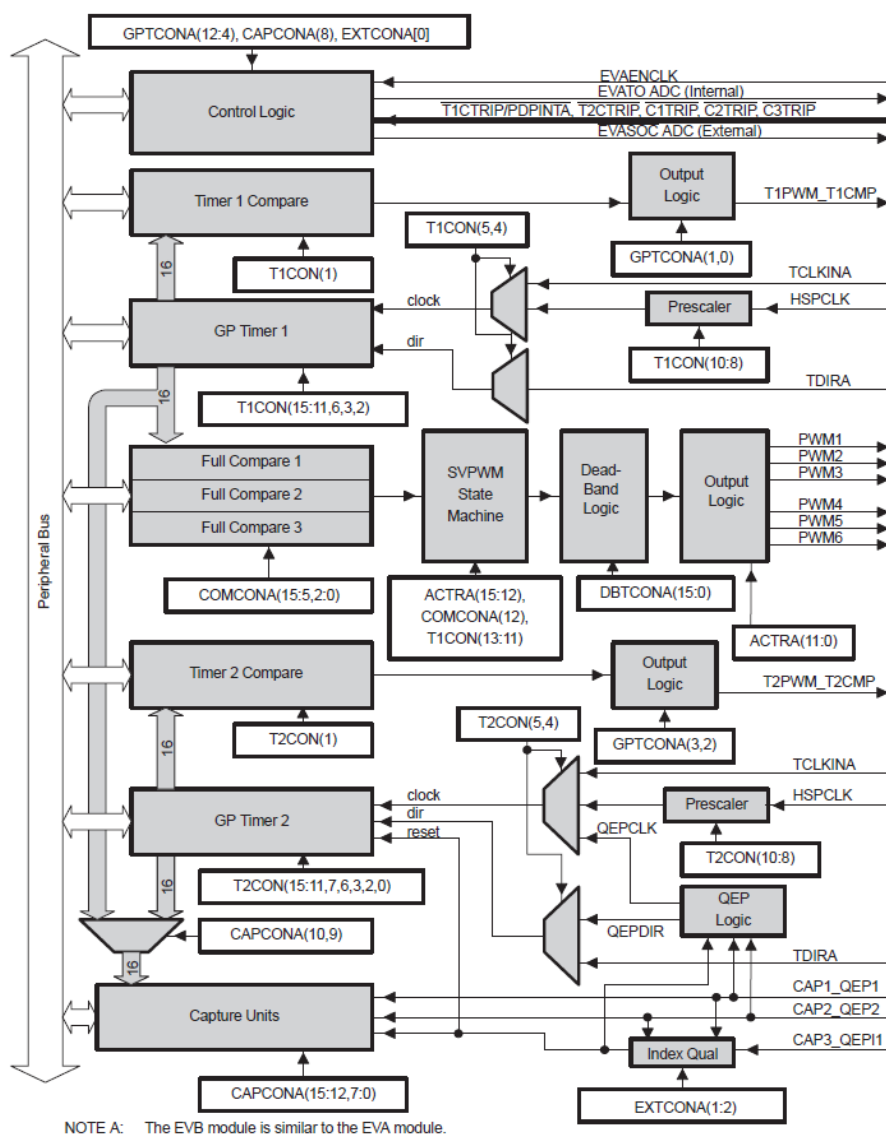
CPU Interrupts	PIE Interrupts							
	INTx.1	INTx.2	INTx.3	INTx.4	INTx.5	INTx.6	INTx.7	INTx.8
INT1	PDPINTA (EV-A)	PDPINTB (EV-B)	reserved	XINT1	XINT2	ADCINT (ADC)	TINT0 (TIMER 0)	WAKEINT (LPM/WD)
INT2	CMP1INT (EV-A)	CMP2INT (EV-A)	CMP3INT (EV-A)	T1PINT (EV-A)	T1CINT (EV-A)	T1UFINT (EV-A)	T1OFINT (EV-A)	reserved
INT3	T2PINT (EV-A)	T2CINT (EV-A)	T2UFINT (EV-A)	T2OFINT (EV-A)	CAPINT1 (EV-A)	CAPINT2 (EV-A)	CAPINT3 (EV-A)	reserved
INT4	CMP4INT (EV-B)	CMP5INT (EV-B)	CMP6INT (EV-B)	T3PINT (EV-B)	T3CINT (EV-B)	T3UFINT (EV-B)	T3OFINT (EV-B)	reserved
INT5	T4PINT (EV-B)	T4CINT (EV-B)	T4UFINT (EV-B)	T4OFINT (EV-B)	CAPINT4 (EV-B)	CAPINT5 (EV-B)	CAPINT6 (EV-B)	reserved
INT6	SPIRXINTA (SPI)	SPITXINTA (SPI)	reserved	reserved	MRINT (McBSP)	MXINT (McBSP)	reserved	reserved
INT7	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
INT8	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
INT9	SCIRXINTA (SCI-A)	SCITXINTA (SCI-A)	SCIRXINTB (SCI-B)	SCITXINTB (SCI-B)	ECAN0INT (CAN)	ECAN1INT (CAN)	reserved	reserved
INT10	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
INT11	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
INT12	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved

**Tab. 1.1: Tabulka vektorů přerušování [9]**

Po provedeném přerušování je nutné vymazat tzv. příznak přerušování, neboť mikroprocesor TMS320F2812 tento příznak nemaže sám po vykonání přerušování.

## 1.4 Event Manager

Mikroprocesor TMS320F2812 obsahuje dva bloky nazývané Event Manager A a Event Manager B, zkráceně EVA a EVB. Oba bloky jsou naprosto stejné, liší se jen v indexech vnitřních bloků. Bloky Event Manager jsou určeny především k PWM modulacím. Celý blok lze vidět na obrázku 1.7, kde jsou také naznačeny jednotlivé registry pro nastavení funkčních bloků.



**Obr. 1.7: Blokové schéma Event Manageru A [9]**

Z obrázku lze vyvodit, že jednotka Event Manager obsahuje dva GP Timery, které se mohou použít jak pro přerušovací systém, tak ke generování PWM signálů. GP Timery jsou pouze 16-ti bitové. Nelze proto dosáhnout tak velkých časových úseků jako je tomu u Timetu 0.

Výstupy GP Timeru směřují nejen na vnitřní sběrnici, ale také do Compare jednotky, ve které se porovnává velikost čítače s nastavenou velikostí. Při komparaci se změní hodnota na výstupu.



## GP TIMER 1

GP Timer 1 je 16-ti bitový časovač se stejnou strukturou jakou mají univerzální časovače Timer 0/1/2. Rozdíl je pouze v jeho větší škále nastavení. Časovače v Event Manager se dají nastavit ve čtyřech čítacích GP modech:

- Stop/Holh Zadržený mód,
- Up counting Nahoru čítací mód,
- Up/Down Řízený nahoru/dolů čítací mód,
- Up/Down Nahoru/dolů průběžný mód.

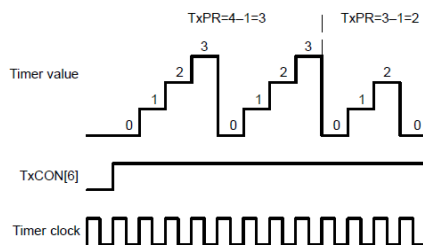
Existuje možnost nastavení hodnoty, do které bude časovač čítat. Hodnota se nastavuje pomocí bitového registru T1PR , jehož velikost je 16 bitů. Pro spuštění časovače je nutné povolit čítání, a to bitem TENABLE v registru T1CON. Jestliže se povolí přerušení časovače, pak lze spustit přerušení až po vykonání jedné periody. Děje se tak proto, že v registru PIEIERx na bitu INTx5 se změní hodnota a vyvolá se daný vektor přerušení. Přerušovací cesta však musí být nastavena až do jádra procesoru. Po vykonání příslušné instrukce musí být vynulovány příznakové bity jak v PIE jednotce, tak v EV Manager jednotce.

### Zadržený mód (Stop/Holh)

Zadržený mód zastaví čítání po jedné periodě.

### Nahoru čítací mód (Up counting)

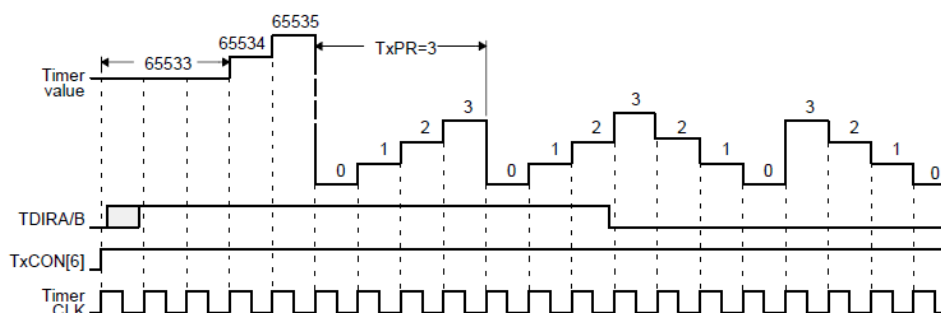
Nahoru čítací mód čítá do hodnoty nastavené v registru TxPR. Poté opět čítá od nuly.



**Obr. 1.8: Nahoru čítací mód [9]**

### Nahoru dolů čítací mód řízený (Up/Down)

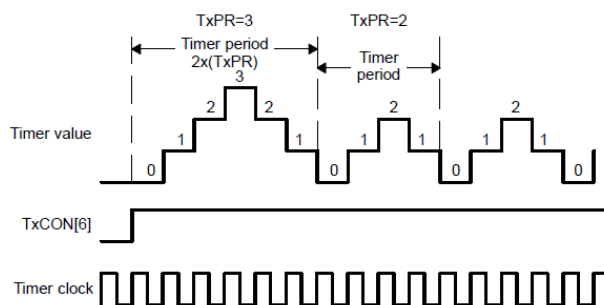
Tento mód je řízený registrem TDIRA/B, který určuje, jestli bude časovač čítat nahoru nebo dolů.



**Obr. 1.9: Nahoru/dolů čítací mód [9]**

### Průběžný mód (Up /Down)

Průběžný mód čítá do hodnoty nastavené v registru T1PR a následně čítá směrem dolů, až do nuly, kde opět začne čítat směrem nahoru.



*Obr. 1.10: Průběžný mód [9]*

### TIMER 1 COMPARE

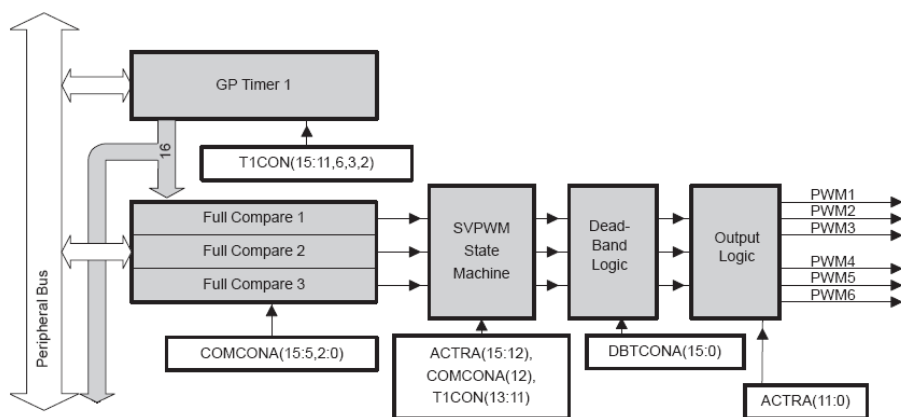
Timer 1 Compace je komparační jednotka porovnávající velikost vstupního signálu z GP Timeru 1 s nastavenou hodnotou v registru T1CMPR, který je rovněž 16-ti bitový. Při komparaci dojde buď k vykonání přerušování, pokud je nastaveno, a nebo ke změně hodnoty na výstupu 1PWM\_T1CMP. Ke změně hodnoty na výstupu dochází také tehdy, když časovač projde nulou. Takto je možné změnou hodnoty komparace řídit spínací úhel u PWM řízení. Ostatní komparační jednotky v EV Manager fungují stejně, jen se nastavují v jiných registrech.

## 1.5 Generování signálů PWM pomocí TMS320F2812

Procesor TMS320F2812 obsahuje dvě Event Manager jednotky EV. Každá EV jednotka je schopná generovat šest pulzů PWM s nastavitelnou ochrannou dobou Dead-Time, dále jeden PWM signál vystupující z Timer1 Compare jednotky označený jako T1PWM\_T1CMP a jeden PWM signál z Timer2 Compare jednotky označený jako T2PWM\_T2CMP. Všechny tyto signály jsou nastavovány a tvořeny v Event Manageru A.

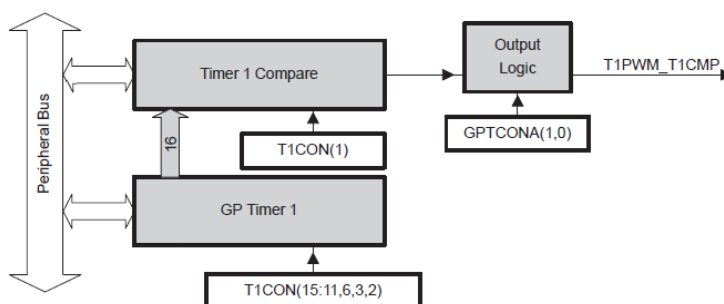
### 1.5.1 Popis tvorby PWM modulace pomocí EV jednotky

Nejprve je nutné spustit a nastavit časovač GP Timer 1. Jeho nastavení již bylo popsáno v kapitole 1.4. Výstupní hodnota časovače je posílána na tři Full Compare jednotky nastavující se registrem COMCONA. Ze všech tří Full Compare jednotek se vytvoří spínací signály. Tyto spínací signály jsou dále posílány do SVPWM jednotky, kde se nastavuje např. vektorová modulace. Po zpracování signálů v SVPWM jednotce se signály posílají do Dead-Band jednotky, v níž se nastavují ochranné doby tzv. Dead-Time. Také se zde vytvoří ze tří signálů signálů šest. Tři se stejnou fází jako vstupní signály a tři inverzní. V Output jednotce jsou následně tyto signály upravovány a posílány na jednotlivé periférie. Na obrázku 1.11 je zobrazeno blokové schéma včetně registrů pro nastavení.



**Obr. 1.11: Bloková struktura PWM v Event Manager A [9]**

Druhá možnost vytvoření PWM signálu je taková, že se opět nejprve nastaví a spustí GP Timer 1 a povolí se Timer 1 Compare jednotka. Z Timer 1 Compare jednotky vychází jeden PWM signál do Output jednotky, kde se nastaví, jaká bude logická hodnota při komparaci. Struktura je zobrazena na obrázku 1.12.



**Obr. 1.12: Bloková struktura PWM v Event Manager A [9]**

## 1.6 Práce s A/D a D/A převodníkem

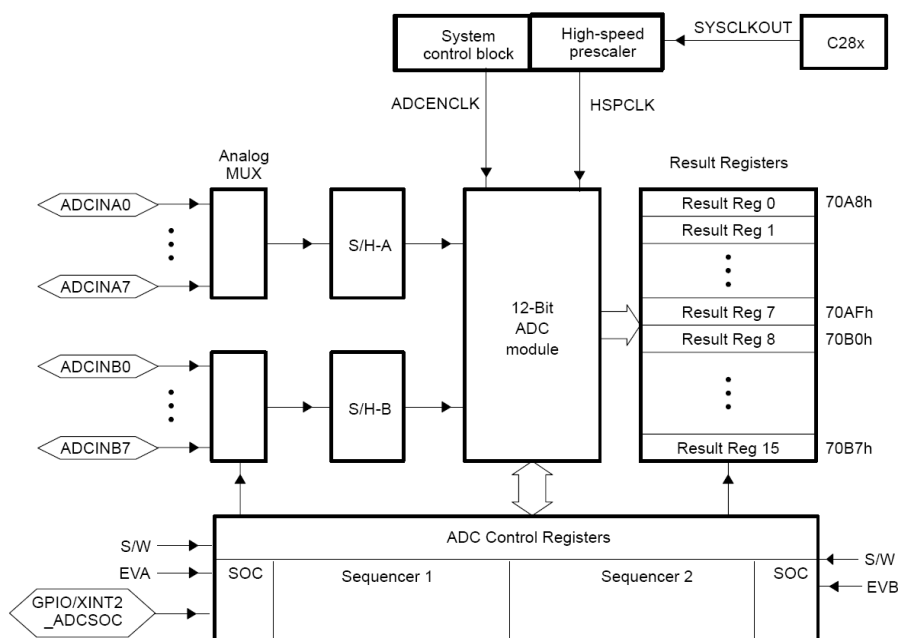
Procesor TMS320F2812 obsahuje jeden 12-ti bitový A/D převodník se 16-ti kanály. Jelikož tento procesor neobsahuje D/A převodník, je třeba D/A převodník připojit externě. D/A převodník je proto umístěn na rozšiřující desce a spojený přes SPI rozhraní.

### 1.6.1 A/D převodník

Základní struktura A/D převodníku procesoru TMS320F2812 je zobrazena na obrázku 1.13. A/D převodník obsahuje 16 kanálů rozdělených na dvě části, část A a část B, po osmi kanálech. Každá část na jeho vstupu je přivedena do multiplexu a následně posílána do jednotky S/H. Z toho důvodu je možné ve stejném čase zachytit hodnotu vstupu jen u jednoho kanálu z části A a u jednoho kanálu z části B. Tyto zachycené signály dále putují do jednoho 12-ti bitového A/D převodníku, v němž se postupně za sebou převedou na digitální hodnotu. Převedené hodnoty se poté ukládají do registru.

### Základní parametry A/D převodníku:

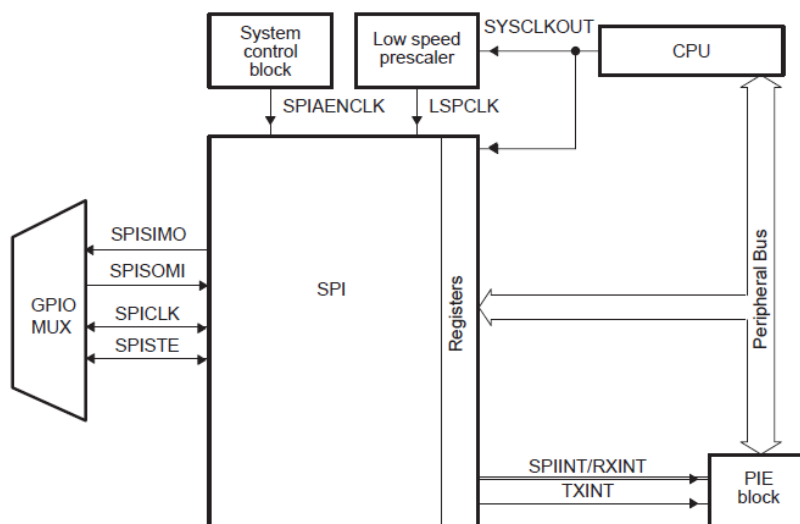
- 12-bit s funkcí S/H,
- analogový vstup 0-3V,
- 16 kanálů ,
- doba převodu  
*200 ns při postupném převodu,*  
*60 ns při využití Pipelined převodu,*
- signály pro začátek převodu  
*Event Manager A,*  
*Event Manager B,*  
*softwarový vstup,*  
*externí vstup z GPIOE1/XINT2\_ADCSOC.*



Obr. 1.13: struktura A/D převodníku [9]

### 1.6.2 Sériové periferní rozhraní (SPI)

SPI je sériové periferní rozhraní určené ke komunikaci mezi samotným procesorem a externími perifériemi, jako je např. komunikace mezi dvěma DSP nebo v tomto případě D/A převodníkem. Komunikace probíhá po čtyřech vodičích, a to SPISIMO, SPISOMI, SPICLK a SPISTE. Nejprve se zařízení domluví, které bude master (tedy řídící) a druhé slave. Poté začne master vysílat hodinový signál určující rychlost komunikace. V tu chvíli je umožněna komunikace mezi zařízeními, kdy master vysílá data po vodiči SPISIMO a přijímá data z vodiče SPISOMI. Mimo jiné lze také nastavit délku přenášeného slova od dvou do 16-ti bitů. Přijímací a vysílací data jsou ukládána do 16-ti úroňové vyrovnávací FIFO paměti.



The SPI module features include:

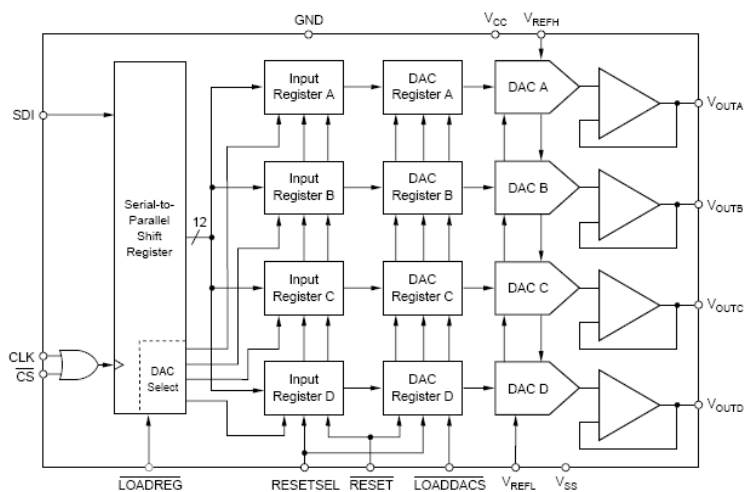
**Obr. č1.14: Bloková struktura zapojení DAC7715 [9]**

### 1.6.3 D/A převodník DAC 7715

Jelikož samotný procesor neobsahuje žádný D/A převodník, je nutné zapojit externí D/A převodník přes SPI výstup. V tomto případě byl použitý D/A převodník typu DAC 7715. Tento převodník je 12-ti bitový se čtyřmi výstupními kanály. Je napájený +15V, -15V. Analogové výstupy mají rozpětí napětí od +10V do -10V a dodají maximální proud  $\pm 5\text{mA}$ . Data jsou přijímána přes sériovou linku z procesoru.

#### Parametry DAC 7715:

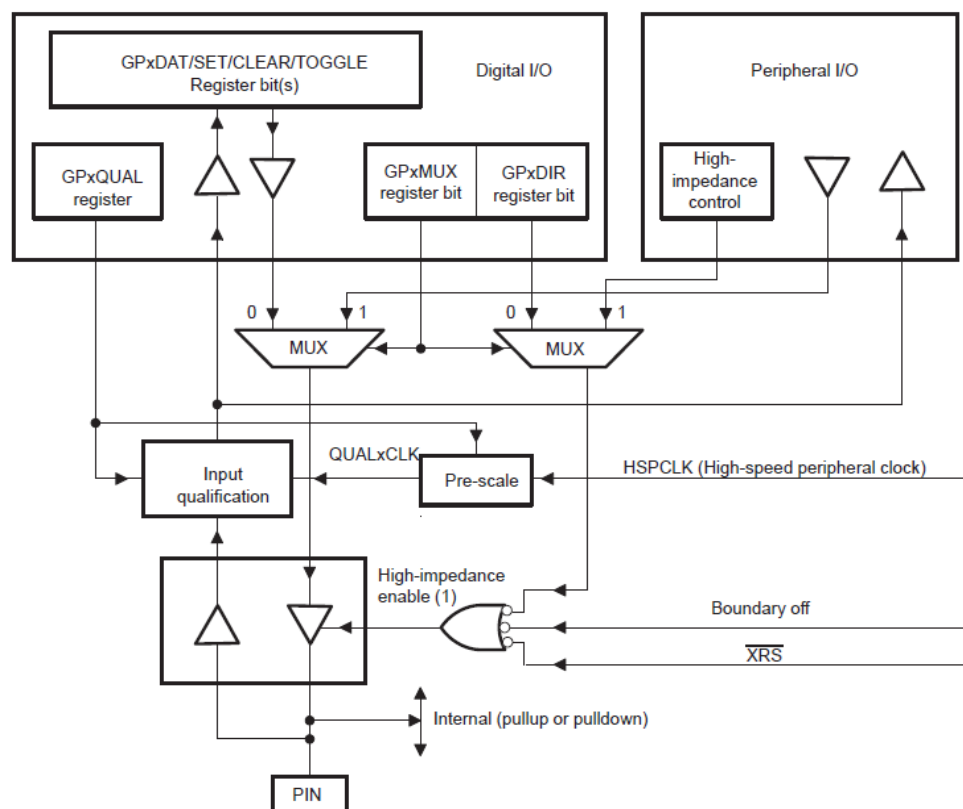
- 12-ti bitový,
- Unipolární nebo Bipolární převod,
- velikost výstupního napětí +10V až -10V,
- délka převodu  $10\mu\text{s}$ .



**Obr. 1.15: Bloková struktura zapojení DAC7715 [12]**

## 1.7 Vstupně-výstupní porty GPIO

Některé výstupní piny mají u procesoru alternativní funkce. Jelikož procesor obsahuje velký počet pinů, je jen přibližně polovina pinů vybavená pouze jednou alternativní funkcí. Převážně jde o funkci výstupu I/O a dále výstup některé z jednotek v procesoru. Pro určení funkce pinu se nastavuje registr GPIO MUX. Registrem GPxDIR lze určit, zda tento pin bude vysílat nebo přijímat. Vše zachycuje blokové schéma na obrázku 1.16.



**Obr. 1.16: Blokové schéma GPIO portů [9]**

## 2 ROZŠÍŘUJÍCÍ DESKA

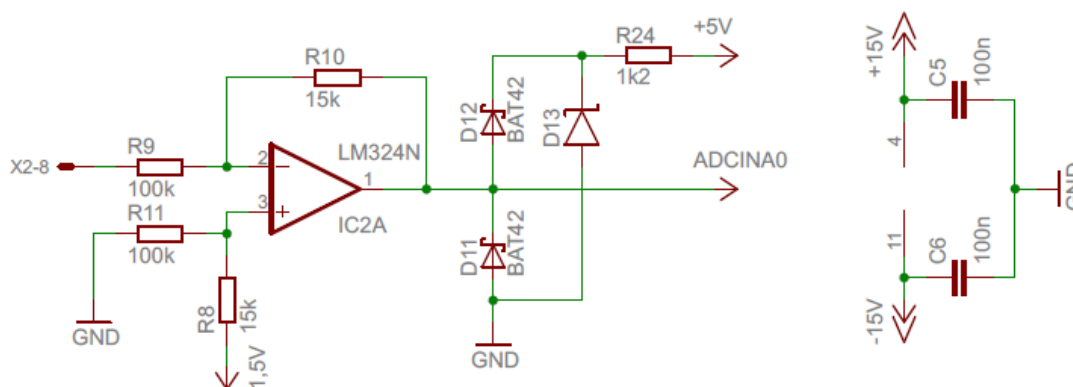
Pro potřeby předmětu Signálové procesory samotný vývojový kit nestačí. Proto je tento kit doplněn o rozšiřující desku, která má samostatné napájení a rozšiřuje možnosti některých periférií.

Mezi parametry rozšiřující desky patří:

- 4 vstupní kanály pro A/D převodník s napětovým rozsahem +10V až -10V,
- 4 výstupní kanály pro D/A převodník s napětovým rozsahem +10V až -10V,
- napájení stejnosměrného napětí +15V, -15V a +9V.

### 2.1 A/D vstup

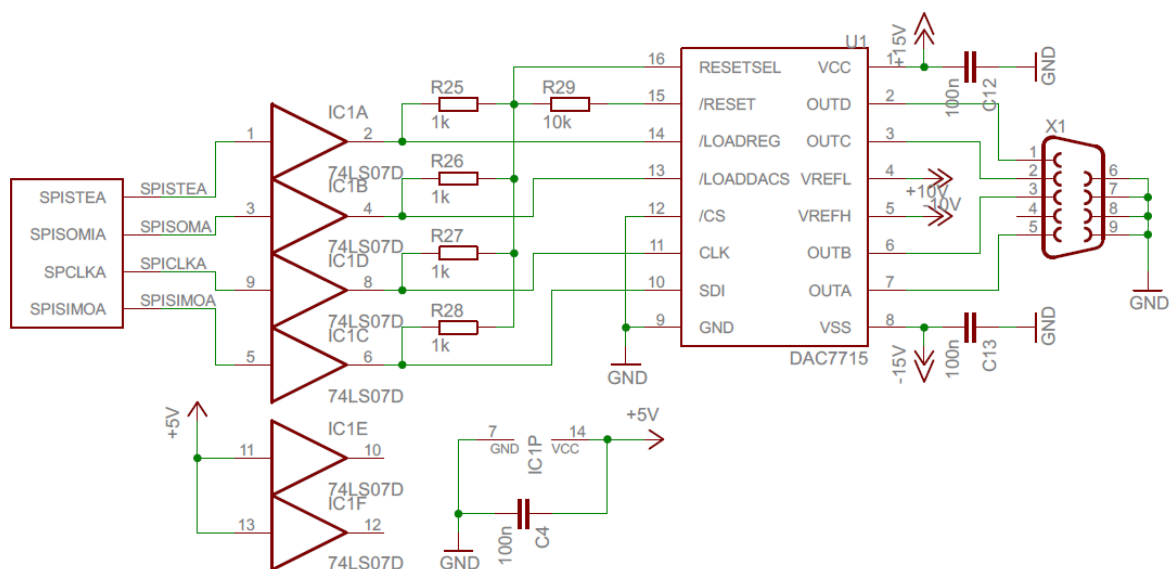
Jedna z nejdůležitějších funkcí na rozšiřující desce je úprava signálů pro A/D převodník nacházející se na procesoru TMS320F2812. A/D převodník má malý napětový rozsah, a to jen od 0V do 3V. K tomu, aby bylo možné měřit signály o velikosti od +10 do -10V, bylo nutné pro úpravu signálu použít invertující zesilovač se zesilením  $A=0,15$  a napětovým posunem +1,5V. Na výstupu zesilovače byly použity lavinové diody pro ochranu proti přepětí. Při zpracování signálů je nutné mít na paměti, že na vstupu je invertující zesilovač a proto výstup bude mít opačnou fázi než vstup. V programu je pak možné tuto fázi obrátit a posunout signál. Přestože procesor obsahuje 16 kanálů A/D převodníku, jsou v tomto případě na rozšiřující desce upraveny jen čtyři kanály (ADCINA0 až ADCINA3).



Obr. 2.1: Zapojení vstupního kanálu A/D

### 2.2 D/A převodník

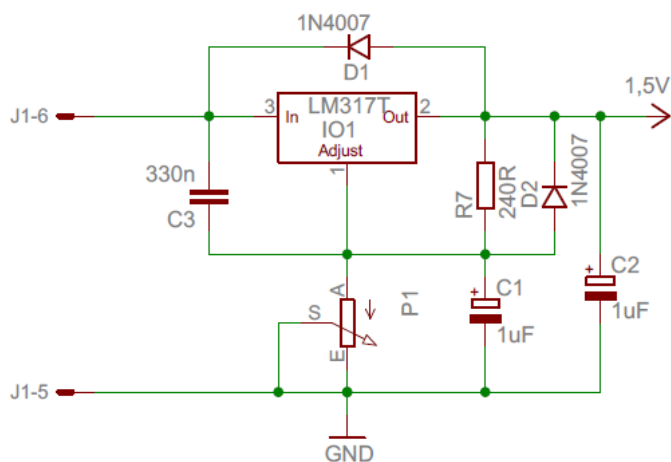
Pokud bude zpracován signál, který bude potřeba dále využít například pro použití filtru, je nutné jej převést z číslicového diskretního signálu do signálu analogového. Samotný procesor a ani vývojový kit neobsahuje D/A převodník a proto ho bylo nutné na rozšiřující desku umístit. K tomuto účelu byl použitý 4-kanálový integrovaný obvod DAC7715 komunikující přes SPI rozhraní.



**Obr. 2.2: Zapojení D/A převodníku**

## 2.3 Napájení

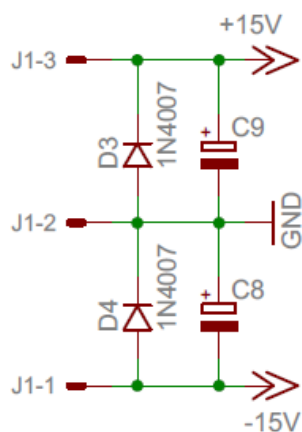
Na desku bylo také nutné umístit napájecí větev na 1,5V pro napěťový posun invertujících zesilovačů. Obvodově je to vyřešeno tak, že z větve 5V se stabilizátorem omezí napětí na 1,5V.



**Obr. 2.3: Stabilizátor na 1,5V**

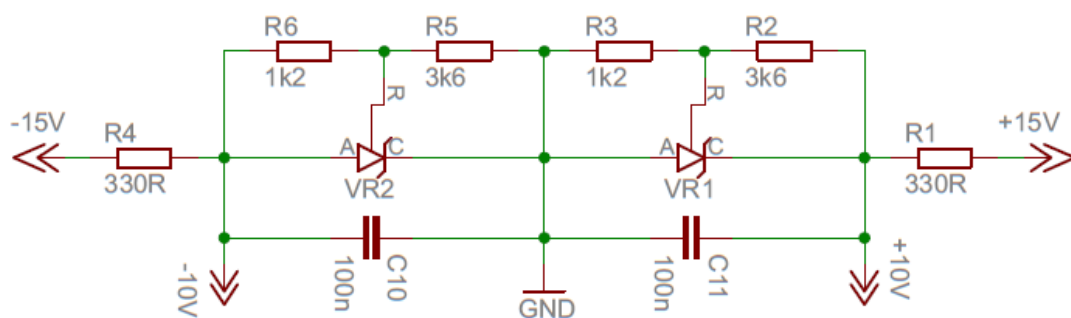
K ochraně proti přepólování jsou na vstupu pro napájení +15V a -15V zapojené dvě diody, které při záměně potenciálů způsobí zkrat. Za těmito diodami jsou ještě další vyhlazovací kondenzátory.





**Obr. 2.4: Vstupní ochrany**

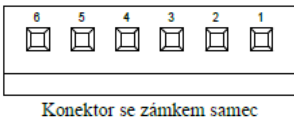
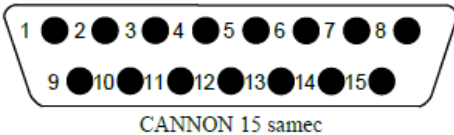
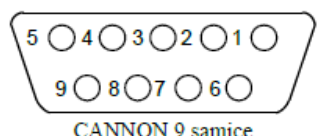
Pro D/A převodník bylo rovněž potřeba vytvořit napětí +10V, -10V. Na desce je to realizováno pomocí odporového děliče.



**Obr. 2.5: Odporový dělič +10V, -10V**

## 2.4 Konektory

Propojení rozšiřující desky a vývojového kitu je realizované pomocí plochých trnů napájených na spodní stranu kitu. Trny se zasouvají do lámací svorkovnice umístěné na rozšiřující desce. Po obvodu rozšiřující desky jsou situovány tři konektory. Význam zapojení těchto konektorů je naznačen v tabulce 2.1

 <p>Konektor se zámkem samec</p>	Napájecí konektor	[1] -15V [2] GND [3] +15V [5] GND [6] +9V
 <p>CANNON 15 samec</p>	Analogový vstup pro A/D převodník	[5] ADCINA0 [6] ADCINA1 [7] ADCINA2 [8] ADCINA3 [10-14] GND
 <p>CANNON 9 samice</p>	Analogový výstup pro D/A převodník	[1] OUTD [2] OUTC [3] OUTB [5] OUTA [6-9] GND

**Tab. 2.1: Zapojení připojovacích konektorů**

## 3 FILTRY

V číslicových soustavách se používají dva typy filtrů, a to filtry s konečnou délkou impulsní odezvy označované jako FIR filtry a filtry s nekonečnou impulsní odezvou označované IIR. Každý typ filtrů má své přednosti i zápory, které budou popsány níže. Jestliže má být filtr realizovatelný, je třeba dodržet podmínky uvedené v následující kapitole.

### 3.1 Kauzalita, linearita a stabilita

Před tím, než budou popsány jednotlivé filtry, je třeba vymezit některé používané pojmy.

Pokud je **systém lineární**, pak vstupní hodnoty  $x_1[n]$  a  $x_2[n]$  mají za následek odezvu  $y_1[n]$  a  $y_2[n]$ . Z uvedeného vyplývá, že odezva na lineární systém je složená ze vstupních signálů

$$x[n] = a \cdot x_1[n] + b \cdot x_2[n], \quad (3.1)$$

$$y[n] = a \cdot y_1[n] + b \cdot y_2[n], \quad (3.2)$$

kde konstanty  $a, b$  jsou libovolná reálná čísla.

Pokud je **systém kauzální**, neboli proveditelný, je nutné, aby impulsní odezva  $h[n]$  byla pro  $n < 0$  nulová. Z toho důvodu nemůže soustava reagovat na změnu dříve, než změna nastane. Matematicky to lze vyjádřit následovně:

$$h[n] = 0, \quad n < 0. \quad (3.3)$$

Kauzální systém má tedy jen pravostrannou pulzní charakteristiku. K ověření kauzality se používá tzv. jednotkový skok. Odezva na tento skok musí být jen pravostranná.

**Systém je stabilní** tehdy, když omezené vstupní signály jsou na výstupu také omezené. Matematicky se to vyjadřuje jako

$$\sum_{n=-\infty}^{\infty} |h[n]| < \infty. \quad (3.4)$$

### 3.2 Z-transformace

K vyšetření přenosu spojité soustavy se používá Laplaceova transformace. Pro vyšetření přenosu diskrétní soustavy je však vhodnější použít Z- transformaci. Z-transformace vychází z Fourierovy transformace a rozšiřuje její použití pro číslicové soustavy. Zjednodušuje řešení lineárních diferenciálních rovnic a umožňuje popis dynamických dějů v diskrétní soustavě.

Z-transformaci vyjadřuje rovnice

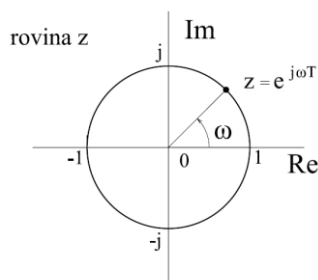
$$X(z) = \sum_{n=0}^{\infty} x[n] \cdot z^{-n}, \quad (3.5)$$

kde  $z$  je komplexní proměnná. Rovnice předpokládá, že posloupnosti jsou jednostranné (nezáporné) a tudíž kauzální, neboli realizovatelné.

Jak již bylo řečeno výše, Z-transformace vychází z Fourierovy transformace, která je definována jako

$$X(e^{j\omega T}) = \sum_{n=0}^{\infty} x[n] \cdot e^{-j\omega T \cdot n}. \quad (3.6)$$

Při porovnání vztahů 3.5 a 3.6 vyplývá, že  $z = e^{j\omega T}$ . Vyjádřením komplexní proměnné v polárním tvaru  $z = r \cdot e^{j\omega T}$  a určením, že vektor  $|z| = 1$ , bude dosaženo tzv. jednotkové kružnice, kde  $\omega T$  je úhel mezi vektorem  $z$  a reálnou osou v radiánech.



**Obr. 3.1: Jednotková kružnice**

U výše uvedené jednotkové kružnice se určuje poloha pólů a nulových bodů filtru. Aby byl filtr stabilní a realizovatelný, musí být póly a nuly v této kružnici, popřípadě na ní.

### 3.3 Diferenční rovnice

Pro matematický popis diskrétní soustavy se používají diferenční rovnice, které jsou ekvivalentem diferenciálních rovnic ve spojité oblasti. Obecně se popisuje vztah mezi vstupem a výstupem diskrétní soustavy následující rovnicí (3.7):

$$y[n] = \sum_{k=0}^M b_k x[n-k] + \sum_{k=1}^M a_k y[n-k], \quad n = 0, 1, 2, \dots \quad (3.7)$$

Po převodu druhého členu, vyskytujícího se na pravé straně, na stranu levou, bude mít rovnice tvar

$$y[n] - \sum_{k=1}^M a_k y[n-k] = \sum_{k=0}^M b_k x[n-k] \quad (3.8)$$

V případě, že bude zapotřebí najít přenosovou funkci, tzv. obraz přenosu, bude nalezena racionální lomená funkce, která se obecně zapisuje vztahem převedeným do  $z$  roviny:

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^M b_k \cdot z^{-k}}{1 - \sum_{k=1}^M a_k \cdot z^{-k}}. \quad (3.9)$$

Tento tvar umožňuje zjistit polohu nulových bodů a pólů obrazu  $H(z)$  v rovině  $z$ . Zároveň je základní rovnicí pro přenos filtrů. S rovnicí bude ještě pracováno v dalších kapitolách této diplomové práce.

### 3.4 Filtry FIR

Číslicové filtry s konečnou délkou impulsní odezvy označované jako FIR (Finite Impulse Response) neobsahují zpětnou vazbu. Lze je popsat diferenciální rovnicí, kde  $x(n)$  označuje vstup a  $y(n)$  výstup. Rovnice má tvar

$$y(n) = b_0 x(n) + b_1 x(n-1) + \dots + b_{N-1} x(n-N+1) = \sum_{k=0}^{N-1} b_k x(n-k), \quad (3.10)$$

kde  $b_0$  až  $b_{N-1}$  jsou reálné koeficienty filtru s délkou  $N$ , řádem  $N-1$  a  $k=0,1,2,\dots,N-1$ . Pokud bude za  $b_k$  dosazeno označení impulzní odezvy  $h(k)$  a provedena  $Z$ -transformace, bude dosazeno této přenosové funkce FIR filtru:

$$H(z) = \sum_{n=0}^{N-1} h(n) z^{-n}. \quad (3.11)$$

Bude-li zapotřebí získat frekvenční charakteristiku, stačí dosadit za  $z=e^{j\omega T}$ . Vznikne přenosová funkce ve tvaru

$$H(e^{j\omega T}) = \sum_{n=0}^{N-1} h(n) e^{-j\omega T n}, \quad (3.12)$$

kde  $\omega T$  je frekvence v radiánech.

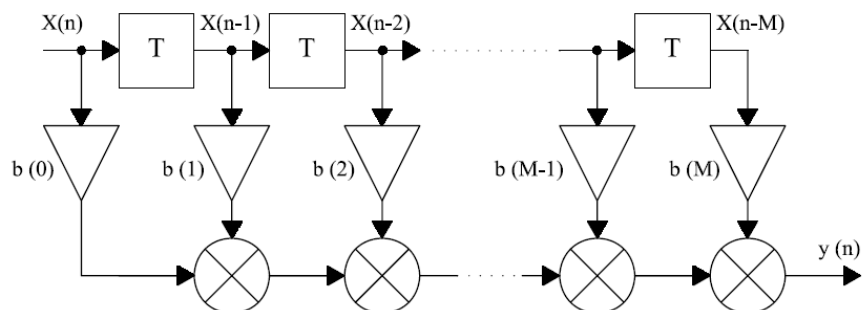
Pro filtr typu FIR neexistuje analogový ekvivalent. Tyto filtry jsou tedy vytvořené jen pro číslicové zpracování. Mají řadu výhodných vlastností. Např.:

- jsou stabilní v celém rozsahu přenosu a proto není zapotřebí zabývat se při návrhu jejich stabilitou,
- velká přesnost bez driftu,
- v celém kmitočtovém pásmu mohou mít lineární fázi,
- jsou jednodušší na výpočet DSP, ale je potřeba větší datová paměť.

Filtry typu FIR mají i své nevýhody.

- používá se vyšší řád,
- jsou nevhodné pro vf signály.

Blokově je filtr zobrazen na obrázku 3.2 a vychází z rovnice 3.14.



**Obr. 3.2: Blokové schéma filtru FIR**

Pro návrh FIR filtru lze použít více základních metod. Jako je např. metoda frekvenčního vzorkování, metoda rovnoměrně zvlněných aproximací a metoda Fourierových řad, která je popsána v kapitole 3.4.2. Ostatními metodami se tato diplomová práce nezabývá. Ti, kteří by se chtěli o uvedených metodách dozvědět více, mohou se o nich dočíst v publikacích. [2], [3], [5].

### 3.4.1 Podmínka lineární fáze filtru

Jednou z nejdůležitějších vlastností filtru FIR je dosažení lineární fáze v celém kmitočtovém rozsahu, což má za následek konstantní zpoždění při různých frekvencích. Aby bylo dosaženo lineární fáze, je potřeba mít symetrickou nebo antisymetrickou impulzní charakteristiku. Pro zajištění uvedených podmínek musí platit

$$h[n] = \pm h[N - n - 1] \quad n = 0, 1, 2, \dots, N - 1 \quad (3.13)$$

kde se pro symetrickou impulzovou charakteristiku používá kladné znaménko před  $h[N - n - 1]$  a pro antisymetrickou impulzovou charakteristiku záporné znaménko před  $h[N - n - 1]$ . Pro délku filtru  $N$  lze určit číslo jak sudé tak i liché. Z toho důvodu mohou nastat čtyři případy impulsové charakteristiky:

- symetrická impulzová charakteristika se sudou délkou filtru  $N$ ,
- symetrická impulzová charakteristika s lichou délkou filtru  $N$ ,
- antisymetrická impulzová charakteristika se sudou délkou filtru  $N$ ,
- antisymetrická impulzová charakteristika s lichou délkou filtru  $N$ .

### Symetrická impulzová charakteristika

Impulzová charakteristika je symetrická podle osy, která prochází polovinou délky filtru  $M = (N-1)/2$ . Jestliže bude délka filtru lichá, pak osa symetrie bude ležet na vzorku  $n=(N-1)/2$ . Pokud bude délka filtru sudá, osa symetrie bude ležet mezi vzorky  $n=(N-2)/2$  a  $n=N/2$ .

### Antisymetrická impulzová charakteristika

Jestliže neprochází fázová charakteristika počátkem, lze hovořit o antisymetrické impulzové odezvě, která má jak konstantní skupinové zpoždění, tak i proměnné fázové zpoždění. Pro osu symetrii u této impulzní charakteristiky při liché délce filtru vždy platí  $h[(N-1)/2]=0$  a při sudé délce filtru  $h[(N-1)/2]=-h[N/2]$ .

V mnoha aplikacích je potřeba mít konstantní skupinové zpoždění tak dlouho, aby se všechny kmitočtové složky v přenášeném signálu dostaly na výstup ve stejný čas.

## 3.4.2 Postup návrhu FIR filtru pomocí Fourierových řad s využitím funkce oken

Tato metoda je v různých literaturách označována jako metoda váhové posloupnosti. Vychází z předpokladu, že frekvenční charakteristika číslicového filtru je periodická s periodou  $2\pi$ . Frekvenční charakteristiku je možné zobrazit jako Fourierovu řadu. Koeficienty Fourierovy řady jsou zároveň koeficienty impulsové charakteristiky. Fourierova řada se vyjadřuje pomocí rovnice

$$h_d[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\omega T}) \cdot e^{j\omega T n} d\omega. \quad (3.14)$$

Po úpravě (3.14) bude mít rovnice tvar

$$h_d[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} |H_d(j\omega)| \cdot [\cos(m\omega) + j \sin(m\omega)] d\omega. \quad (3.15)$$

kde  $m=n-(N-1)/2$

Úpravou rovnice bude získána rovnice pro výpočet koeficientů filtru. Za meze integrálu bude dosazeno propustné pásmo převedené do kmitočtové charakteristiky. Například, pro dolní propust budou meze od 0 do  $\omega_d$  a pro horní propust od  $\omega_h$  do 0.

Pro upřesnění je ještě třeba uvést, co zde znamená úhlová rychlost. Úhlovou rychlostí  $\omega$  je myšlen poměr mezní frekvence k frekvenci vzorkování vynásobené  $2\pi$ :

$$\omega_d = 2\pi \frac{f_d}{f_v}, \quad \omega_h = 2\pi \frac{f_h}{f_v}, \quad (3.16)$$

kde  $\omega_d$  je dolní kmitočet,  $\omega_h$  horní kmitočet,  $f_d$  spodní frekvence,  $f_h$  horní frekvence a  $f_v$  vzorkovací frekvence.

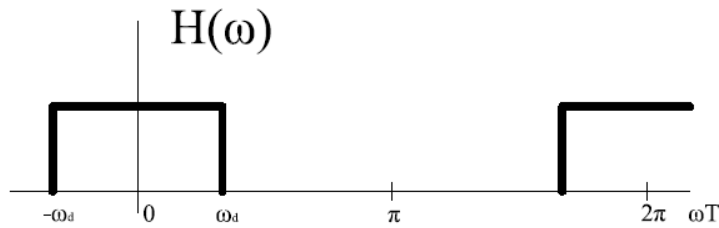


Pokud bude zvolena symetrická impulsní charakteristika s lichým počtem koeficientů, je sinus rovnice (3.16) nulová. Pro výpočet tak bude dosažena rovnice

$$h_d[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \cos(m\omega) d\omega. \quad (3.17)$$

Dále je ukázáno, jak se po integrování vypočtou dané koeficienty a také jak se vypočte koeficient, pokud je  $m=0$ . Pro  $m=0$  se po integraci používá l'Hospitalova pravidla, které derivuje čitatele i jmenovatele podle  $m$ .

### Dolní propust

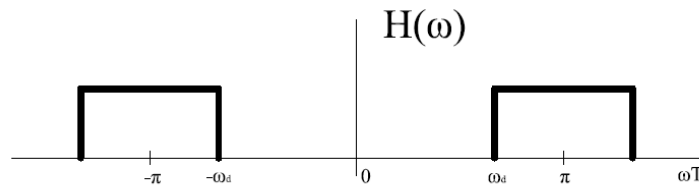


**Obr. 3.3: Kmitočtová charakteristika dolní propusti**

$$h_d[n] = \frac{1}{2\pi} \int_0^{\omega_d} [\cos(m\omega) + j \sin(m\omega)] d\omega \quad (3.18)$$

$$h_d[n] = \begin{cases} \frac{\omega_d}{\pi} & \text{prom} = 0 \\ \frac{\sin(\omega_d m)}{m\pi} & \text{prom} \neq 0 \end{cases} \quad (3.19)$$

### Horní propust

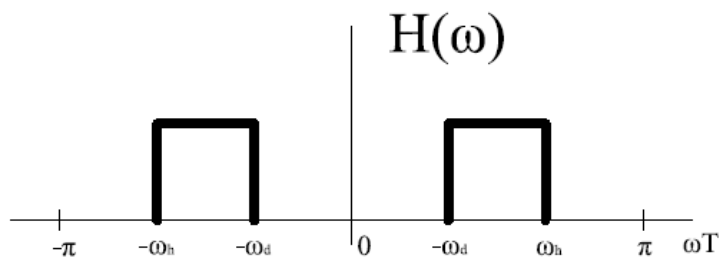


**Obr. 3.4: Kmitočtová charakteristika horní propusti**

$$h_d[n] = \frac{1}{2\pi} \int_{\omega_h}^0 [\cos(m\omega) + j \sin(m\omega)] d\omega \quad (3.20)$$

$$h_d[n] = \begin{cases} 1 - \frac{\omega_h}{\pi} & \text{prom} = 0 \\ -\frac{\sin(\omega_h m)}{m\pi} & \text{prom} \neq 0 \end{cases} \quad (3.21)$$

## Pásmová zádrž

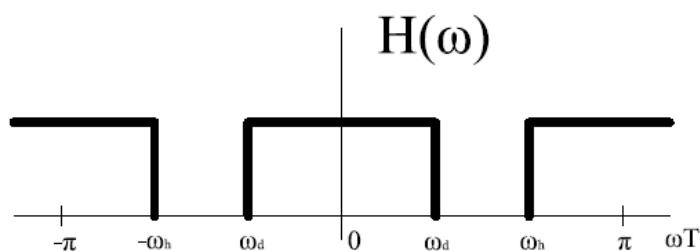


**Obr. 3.5: Kmitočtová charakteristika pásmové propusti**

$$h_d[n] = \frac{1}{2\pi} \int_{\omega_h}^{\omega_d} [\cos(m\omega) + j \sin(m\omega)] d\omega \quad (3.22)$$

$$h_d[n] = \begin{cases} 1 - \frac{\omega_h - \omega_d}{\pi} & \text{prom } m = 0 \\ \frac{1}{m\pi} [\sin(\omega_d m) - \sin(\omega_h m\pi)] & \text{prom } m \neq 0 \end{cases} \quad (3.23)$$

## Pásmová propust



**Obr. 3.6: Kmitočtová charakteristika pásmové zádrže**

$$h_d[n] = \frac{1}{2\pi} \int_{\omega_d}^{\omega_h} [\cos(m\omega) + j \sin(m\omega)] d\omega \quad (3.24)$$

$$h_d[n] = \begin{cases} \frac{\omega_h - \omega_d}{\pi} & \text{prom } m = 0 \\ \frac{1}{m\pi} [\sin(\omega_h m) - \sin(\omega_d m)] & \text{prom } m \neq 0 \end{cases} \quad (3.25)$$

### 3.4.3 Okna používaná pro návrh filtru FIR

Při navrhování číslicových filtrů FIR se vychází z ideální frekvenční charakteristiky  $H_d(e^{j\omega T})$ , kde je přenosová funkce definovaná v intervalu  $(-\pi, \pi)$  a je periodická s periodou  $2\pi$ . Za předpokladu, že bude třeba nalézt impulzní odezvu zadaného filtru pomocí Fourierovy transformace diskrétního signálu, bude zjištěno, že impulzní odezva je nekonečná a nekauzální. Z tohoto důvodu je daný filtr nerealizovatelný. K použití impulzní odezvy daného filtru je nutné užít tzv. okna. Okna jsou konečně

definované funkce o délce  $N$ , které se násobí s impulzovou odezvou. Takto bude dosaženo nejen impulzové odezvy s konečným počtem prvků  $N$ , ale i konečný počet koeficientu pro daný filtr.

Pokud z nekonečného systému bude získán systém konečný, zhoršuje se frekvenční charakteristika vlivem konvoluce spektra okna a požadované frekvenční charakteristiky. Aby se tento jev tolik neprojevoval, vynásobíme impulzovou charakteristiku tzv. okny, podle vztahu

$$h_w[n] = h_d[n] \cdot w[n] \quad (3.26)$$

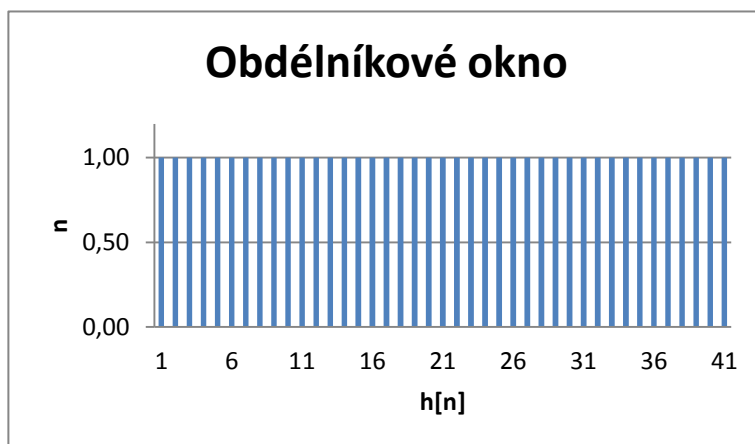
Používá se několik typů oken, jejichž cílem je omezit zvlnění amplitudové charakteristiky  $H(\omega)$ , tzv. Gibbsův jev.

Pro realizovatelnost filtru je ještě nutné posunout impulsní charakteristiky o  $M=(N-1)/2$  vzorků. Posunutí impulzové charakteristiky poskytuje lineární fáze.

### 3.4.3.1 Obdélníkové okno

Obdélníkové okno je nejjednodušší typ okna a proto má také nejhorší vlastnosti. Je to jen vyseknutý úsek signálu bez úpravy hodnot. Obdélníkové okno je definované vztahem (3.27)

$$w[n] = \begin{cases} 1 & \text{pro } 0 \leq n \leq N-1 \\ 0 & \text{pro ostatní } n \end{cases} \quad (3.27)$$

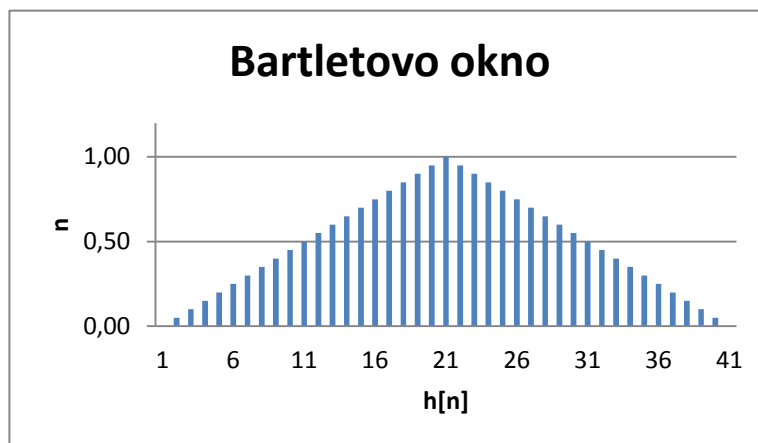


Obr. 3.7: Obdélníkové okno s délkou filtru  $N = 41$

### 3.4.3.2 Bartletovo okno

Bartletovo okno má pozvolné přechody mezi úrovněmi. Gibbsův jev je zde menší než u obdélníkového okna. U tohoto typu je okno trojúhelníkové. Rovnice Bartletova okna má tvar

$$w[n] = \begin{cases} \frac{2n}{N-1} & \text{pro } 0 \leq n \leq \frac{N-1}{2} \\ 2 - \frac{2n}{N-1} & \text{pro } \frac{N-1}{2} \leq n \leq N-1 \\ 0 & \text{pro ostatní } n \end{cases} \quad (3.28)$$

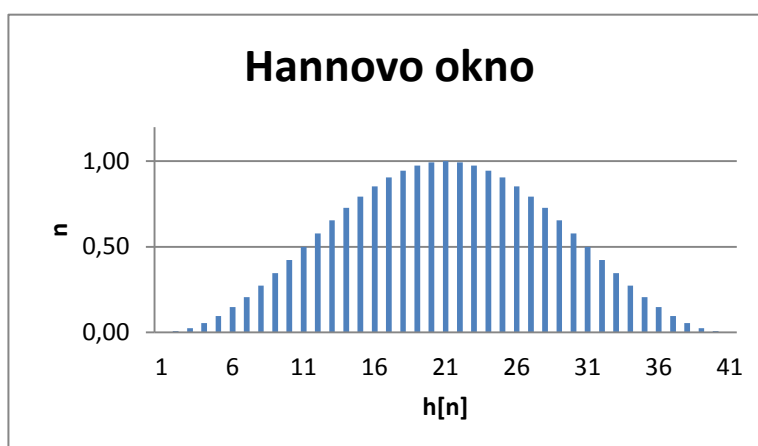


**Obr.3.8: Bartletovo okno s délkou filtru  $N = 41$**

### 3.4.3.3 Hannovo okno

Hannovo okno může být také označováno jako Hanning. Je to typ okna cosinové. Hannovo okno je dáno vztahem

$$w[n] = \begin{cases} 0,5 \left[ 1 - \cos\left(\frac{2\pi n}{N-1}\right) \right] & \text{pro } 0 \leq n \leq N-1 \\ 0 & \text{pro ostatní } n \end{cases} \quad (3.29)$$

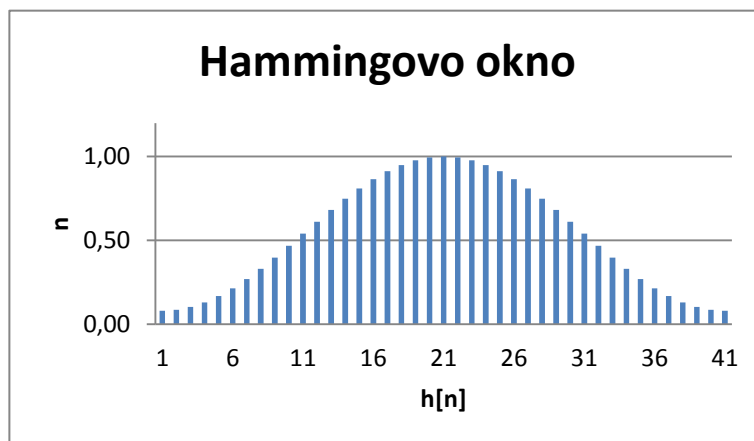


**Obr.3.9: Hannovo okno s délkou filtru  $N = 41$**

### 3.4.3.4 Hammingovo okno

Hammingovo okno je rovněž typu cosinus, ale obsahuje malé nespojitosti na okrajích v časové funkci. Toto okno lze popsat následovně:

$$w[n] = \begin{cases} 0,54 - 0,46 \cdot \cos\left(\frac{2\pi n}{N-1}\right) & \text{pro } 0 \leq n \leq N-1 \\ 0 & \text{pro ostatní } n \end{cases} \quad (3.30)$$

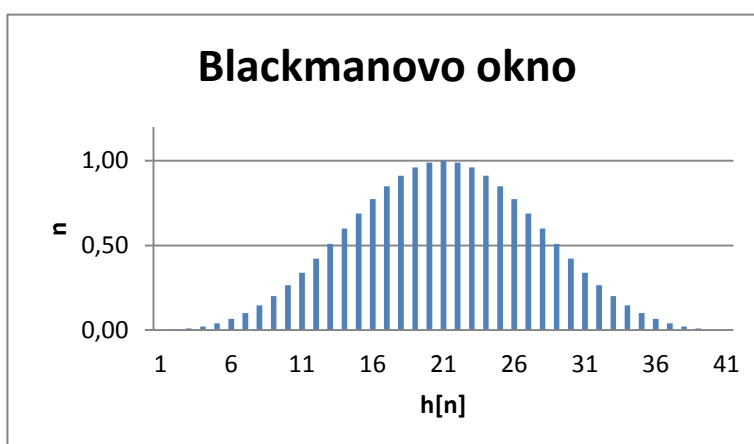


Obr. 3.10: Hammingovo okno s délkou filtru  $N = 41$

### 3.4.3.5 Blackmanovo okno

Blackmanovo okno je také typu cosinus. Od předchozích oken se liší tím, že obsahuje člen s druhou harmonickou složkou. Rovnice popisující toto okno se vyjadřuje vztahem

$$w[n] = \begin{cases} 0,42 - 0,5 \cdot \cos\left(\frac{2\pi n}{N-1}\right) + 0,08 \cdot \cos\left(\frac{4\pi n}{N-1}\right) & \text{pro } 0 \leq n \leq N-1 \\ 0 & \text{pro ostatní } n \end{cases} \quad (3.31)$$



Obr. 3.11: Blackmanovo okno s délkou filtru  $N = 40$

### 3.4.3.6 Srovnání oken

Níže uvedená tabulka srovnává jednotlivá okna, která byla popsána v předchozích kapitolách. Porovnávána je šířka přechodového pásma a útlum v nepropustném pásmu.

Typ okna	Šířka pásma	Min.odstup
Obdélníkové	$1.8\pi/N$	21 dB
Bartlett	$6.1\pi/N$	25 dB
Hann	$6.2\pi/N$	44 dB
Hamming	$6.6\pi/N$	53 dB
Blackman	$11\pi/N$	74 dB

## 3.5 Filtry IIR

Filtr IIR obsahuje ve své struktuře zpětnovazební člen a proto je označován jako filtr s nekonečně dlouhou impulzovou odezvou IIR (Infinite Impulse Response). Tyto filtry vycházejí z ekvivalentu pro analogové filtry, kde přenosová funkce analogového filtru  $H(p)$  se pomocí transformace převede do číslicového tvaru  $H(z)$ .

Vzhledem k tomu, že filtr typu IIR má zpětnovazební člen přenášející výstupní hodnotu ( $y[n]$ ) na vstup filtru, je jeho rovnice přenosu racionálně lomenou funkcí

$$H(z) = \frac{\sum_{i=0}^M b_i z^{-i}}{1 + \sum_{i=1}^N a_i z^{-i}}. \quad (3.32)$$

Filtr IIR má v přenosové funkci nuly i póly, což má za následek, že u tohoto filtru hraje významnou roli otázka stability. Jelikož se budou frekvenční požadavky aproximovat podílem polynomů, bude také řád filtru IIR nižší a frekvenční charakteristika lepší. Naopak to přináší komplikace při návrhu a výskyt nežádoucích jevů při implementaci. Pro zápis v časové oblasti použijeme diferenční rovnici

$$y[n] = \sum_{i=0}^M b_i x[n-i] - \sum_{i=1}^N a_i x[n-i]. \quad (3.33)$$

První suma je stejná jako u filtru FIR, avšak druhá suma rovnice je zpětnovazební (rekurzivní) člen způsobující nekonečné trvání impulzní odezvy.

### 3.5.1 Metody návrhu číslicových filtrů IIR

Pro návrh číslicových filtrů typu IIR se metody dělí na dvě skupiny:

- **metody využívající analogových prototypů**, které se používají nejčastěji. Jak již ze samotného názvu vyplývá, tyto metody využívají analogové prototypy filtrů. Patří zde bilineární transformace nebo vlnové číslicové filtry.
- **metody přímého návrhu v rovině  $z$** . Tato metoda je sice přesnější, ale pro její výpočtovou náročnost méně používaná. Patří sem např. metoda nejmenších čtverců a identifikační parametrické metody.

Tato diplomová práce se zaměřuje pouze na metodu bilineární transformace, o které bude pojednáno v následující kapitole.

### 3.5.2 Bilineární transformace

Bilineární transformace se používá tehdy, když je nutné převést obrazy Laplaceovy transformace roviny  $p$  na obraz transformace roviny  $z$ . Souvislost mezi těmito rovinami vyjadřuje vztah 3.34, kde  $T$  je perioda vzorkování:

$$z = e^{p \cdot T}. \quad (3.34)$$

Pokud se musí filtr z roviny  $p$  převést do roviny  $z$ , vyjádří se proměnná  $p$  následovně:

$$p = \frac{1}{T} \ln z = \frac{1}{T} \frac{2(z-1)}{z+1 - \frac{(z-1)^4}{3(z+1)(z-1)^2}} \quad (3.35).$$

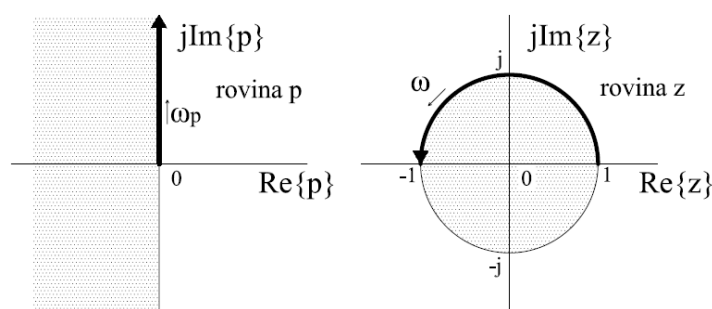
Z logaritmické funkce se použije jen první člen rozvoje řetězové funkce. Obecně lze tedy říci, že pro převod z jedné roviny do druhé stačí znát převodní vztah

$$p \rightarrow \frac{2}{T} \cdot \frac{z-1}{z+1} \quad (3.36)$$

a opačně

$$z \rightarrow \frac{\frac{2}{T} + p}{\frac{2}{T} - p} \quad (3.37)$$

Na obrázku je pak vidět, jak se jednotlivé roviny převedou.



**Obr. 3.12: Transformace z roviny  $p$  do roviny  $z$**

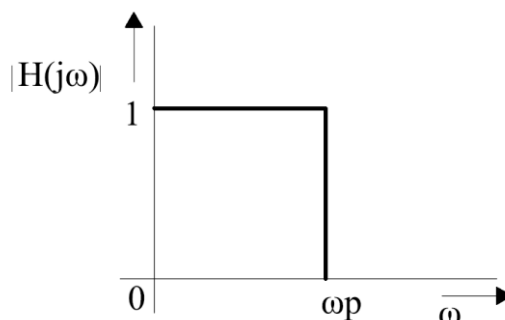
Pro stabilitu filtru IIR je potřeba zajistit, aby všechny body v rovině  $p$  byly v levé části. Po převodu do roviny  $z$  musí ležet v kružnici a nebo na ní. Při této transformaci samozřejmě nelze zapomenout na převod kmitočtových os. Převádí se kmitočtová osa, která je definovaná od  $-\infty$  do  $+\infty$ , na kmitočtovou osu s definicí od 0 do  $\pi$ . Pouze takto lze převádět filtry DP, HP, PZ, PP. Pro převod do číslicové transformace se pak používá vztah

$$\omega_z = \frac{2}{T} \operatorname{tg} \frac{\omega T}{2} \quad (3.38)$$

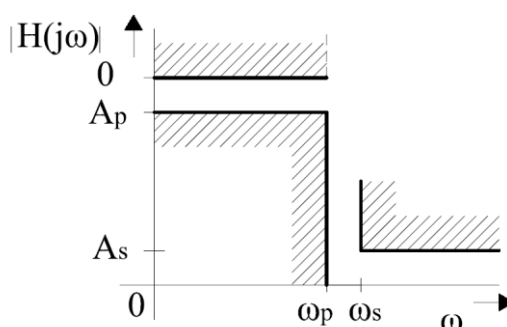
nebo při opačné transformaci

$$\omega = \frac{2}{T} \operatorname{arctg} \frac{\omega_z T}{2}. \quad (3.39)$$

Základem pro návrh všech filtrů je ideální charakteristika dolní propusti. Tato tzv. ideální dolní propust se převede na tzv. normovanou dolní propust (NDP), která už není ideální, neboť obsahuje v propustném pásmu určitou chybu přechodové charakteristiky. V přechodovém pásmu normované dolní propusti přechodová charakteristika klesá. Nakonec je v nepropustném pásmu pásmo, ve kterém je možné zvlnění.



**Obr. 3.13: Přechodová charakteristika ideální dolní propusti**



**Obr. 3.14: Toleranční schéma dolní propusti**

Z normované dolní propusti vycházejí tyto aproximace:

- Butterworthova aproximace,
- Čebyševová aproximace,
- Inverzní Čebyševová aproximace,
- Eliptická aproximace.

Vyjmenované aproximace se liší jen v průběhu přechodové charakteristiky. Postupně budou vysvětleny v následujících kapitolách této diplomové práce.

Jak již bylo řečeno, veškeré typy filtrů, jako je DP, HP, PZ, PP, vycházejí z rovnice pro normovanou dolní propust. Do rovnice se za parametr  $p$  dosadí příslušný výraz, který se převede



normovanou dolní propustí na požadovaný filtr. Následující tabulka ukazuje postup jak převést normovanou dolní propust na ostatní filtry.

Přechod z NDP na	Převodní vztahy	Parametry
Dolní propust	$p \rightarrow \frac{p}{\omega_p}$	$\omega_p$ – mezní úhlový kmitočet propustného pásma DP
Horní propust	$p \rightarrow \frac{\omega_p}{p}$	$\omega_p$ – mezní úhlový kmitočet propustného pásma HP
Pásmová propust	$p \rightarrow \frac{p^2 + \omega_0^2}{p\omega_\Delta} = \frac{p^2 + \omega_{p1} \cdot \omega_{p2}}{p\omega_{p2} \cdot \omega_{p1}}$	$\omega_{p1}, \omega_{p2}$ – hranice propustného pásma PP
Pásmová zadrž	$p \rightarrow \frac{p\omega_\Delta}{p^2 + \omega_0^2} = \frac{p\omega_{n2} \cdot \omega_{n1}}{p^2 + \omega_{n1} \cdot \omega_{n2}}$	$\omega_{n1}, \omega_{n2}$ – hranice nepropustného pásma PZ

**Tab. 3.1: Kmitočtové transformace**

### 3.5.3 Butterworthova aproximace

Pro Butterworthovou aproximaci je typický monotónní průběh bez převýšení. Nevýhodou je však velké přenosové pásmo. Ke zmenšení přechodového pásma je třeba použít větší řád filtru. Aproximační vztah pro Butterworthovou dolní propust je ve spojitě oblasti popsán rovnicí

$$|H(j\omega)|^2 = \frac{1}{1 + \left(\frac{\omega}{\omega_c}\right)^{2N}} = \frac{1}{1 + \varepsilon^2 \left(\frac{\omega}{\omega_p}\right)^{2N}}, \quad (3.40)$$

kde  $\omega_c$  je mezní frekvence propustného pásma, při poklesu o 3 dB,  $\omega_p$  je mezní frekvence propustného pásma při  $|H(j\omega)|^2 = 1/(1+\varepsilon^2)$  a N je řád filtru.

Pro návrh filtru je velmi důležitá velikost řádu filtru, která se vypočítá podle vztahu

$$N \geq \frac{1}{2} \frac{\log \left[ \frac{(1/A_s^2) - 1}{(1/A_p^2) - 1} \right]}{\log \left( \frac{\omega_s}{\omega_p} \right)}, \quad (3.41)$$

kdy mezní frekvence  $\omega_c$  je dána rovnicí

$$\omega_c = \frac{\omega_p}{\sqrt[2N]{\left(\frac{1}{A_p}\right)^2 - 1}}. \quad (3.42)$$

### 3.5.4 Čebyševová aproximace

Pro Čebyševovu aproximaci je typické zvlnění v propustném pásmu a hladký monotónní průběh v nepropustném pásmu. Její přechodová charakteristika je mnohem strmější:

$$|H(j\omega)|^2 = \frac{1}{1 + \varepsilon^2 \cdot C_N^2\left(\frac{j\omega}{j\omega_c}\right)}, \quad (3.43)$$

kde  $\varepsilon$  je parametr udávající velikost přechodové charakteristiky v propustném pásmu a  $C_N^2(x)$  je Čebyševův polynom řádu  $N$

### 3.5.5 Inverzní Čebyševová aproximace

Opakem průběhu modulové charakteristiky je inverzní Čebyševová aproximace, u které bude v propustném pásmu funkce monotónní a v nepropustném pásmu funkce zvlněná. Inverzní Čebyševová aproximace je dána vztahem

$$|H(j\omega)|^2 = \frac{1}{1 + \left[ \varepsilon^2 \cdot C_N^2\left(\frac{j\omega}{j\omega_c}\right) \right]^{-1}} = \frac{\varepsilon^2 \cdot C_N^2\left(\frac{\omega_c}{\omega}\right)}{1 + \varepsilon^2 \cdot C_N^2\left(\frac{\omega_c}{\omega}\right)}. \quad (3.44)$$

### 3.5.6 Eliptická aproximace

U Eliptické aproximace bude v přechodové charakteristice dosaženo velké strmosti, ale funkce bude zvlněná jak v propustném, tak nepropustném pásmu:

$$|H(j\omega)|^2 = \frac{1}{1 + \varepsilon^2 \cdot U_N^2(\omega)}. \quad (3.45)$$

### 3.5.7 Postup návrhu IIR filtru

K realizaci filtru se často používá kaskádní forma. Při její realizaci se vychází ze součinu filtru maximálně druhého řádu. Pro lepší názornost jsou uvedeny rovnice pro kaskádní řazení Butterworthova filtru:

$$H(p) = \prod_{k=1}^{N/2} \frac{B_k \omega_c^2}{p^2 + b_k \omega_c p + c_k \omega_c^2}, \quad N = 2, 4, 6, \dots, \quad (3.46)$$

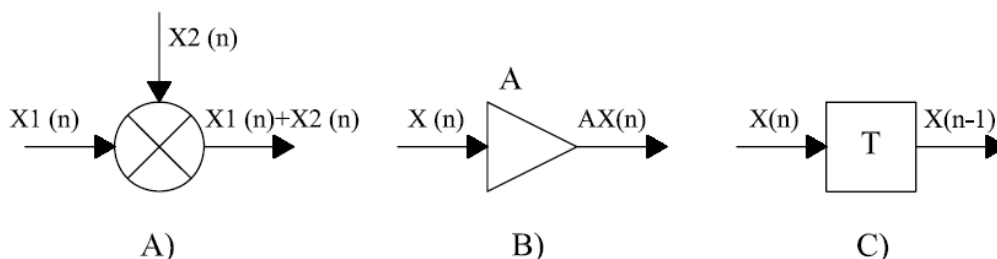
$$H(p) = \frac{B_0 \omega_c}{p + c_0 \omega_c} \prod_{k=1}^{(N-1)/2} \frac{B_k \omega_c^2}{p^2 + b_k \omega_c p + c_k \omega_c^2}, \quad N = 3, 5, 7, \dots, \quad (3.47)$$

$$b_k = 2 \sin \left[ \frac{(2k-1)\pi}{2N} \right], \quad (3.48)$$

$$c_k = 1. \quad (3.49)$$

# 4 ALGORITMY A ZPRACOVÁNÍ JEDNOTLIVÝCH FILTRŮ

Číslicové filtry se dají poskládat ze tří základních bloků. Sčítačka, násobení konstantou a jednotkové zpoždění.



*Obr. 4.1: A) sčítačka, B) násobení konstantou, C) jednotkové zpoždění*

Výše uvedené operace jsou matematicky vyjadřovány v diskretní soustavě takto:

- součet dvou signálů  $x_1[n]$  a  $x_2[n]$ :

$$y[n] = x_1[n] + x_2[n], \quad n = 0, 1, 2, \dots, \quad (4.1)$$

- součin signálu  $x[n]$  s konstantou  $A$ :

$$y[n] = A \cdot x[n], \quad n = 0, 1, 2, \dots, \quad (4.2)$$

- zpoždění signálů  $x[n]$  o vzorkovací krok  $k$ :

$$y[n] = x[n - k], \quad n = 0, 1, 2, \dots, \quad k = 1, 2, \dots, M. \quad (4.3)$$

## 4.1 Vyjádření čísla v pevné řádové čárce

K vyjádření tvaru desetinného čísla se používá v procesorech s pevnou řádovou čárkou tzv. Q formát čísla. Ten určuje, jaký bude počet bitů za desetinou čárkou. Desetinné číslo se vypočte tak, že se vynásobí příslušným váhovým koeficientem. Tento váhový koeficient se zvolí podle hodnoty čísla před desetinou čárkou. Někdy je číslo označované jako např. Q1.15, kde jednička udává počet bitů před desetinou čárkou. V této diplomové práci jsou tyto převody ukázané v kapitole 5.

### Příklad:

Číslo 3,6543 je potřeba převést na Q formát. Podle tabulky se určí, že spadá do rozsahu 7,999755859375. Následně se toto číslo vynásobí váhovým koeficientem 4096.

$$Q_{12} = 3,6543 \cdot 4096 = \underline{14963} \quad (4.4)$$

Formát Qn	Váhový koeficient	Rozsah v šestnáctkové a desítkové číselné soustavě
Q15	$2^{15}=32768$	$(7FFF)_{16}=(0,999969482421875)_{10}$ až $(8000)_{16}=(-1)_{10}$
Q14	$2^{14}=16384$	$(7FFF)_{16}=(1,99993896484375)_{10}$ až $(8000)_{16}=(-2)_{10}$
Q13	$2^{13}=8192$	$(7FFF)_{16}=(3,9998779296875)_{10}$ až $(8000)_{16}=(-4)_{10}$
Q12	$2^{12}=4096$	$(7FFF)_{16}=(7,999755859375)_{10}$ až $(8000)_{16}=(-8)_{10}$
Q11	$2^{11}=2048$	$(7FFF)_{16}=(15,99951171875)_{10}$ až $(8000)_{16}=(-16)_{10}$
Q10	$2^{10}=1024$	$(7FFF)_{16}=(31,99902234375)_{10}$ až $(8000)_{16}=(-32)_{10}$
Q9	$2^9=512$	$(7FFF)_{16}=(63,998046875)_{10}$ až $(8000)_{16}=(-64)_{10}$
Q8	$2^8=256$	$(7FFF)_{16}=(127,99609375)_{10}$ až $(8000)_{16}=(-128)_{10}$
Q7	$2^7=128$	$(7FFF)_{16}=(255,9921875)_{10}$ až $(8000)_{16}=(-256)_{10}$
Q6	$2^6=64$	$(7FFF)_{16}=(511,984375)_{10}$ až $(8000)_{16}=(-512)_{10}$
Q5	$2^5=32$	$(7FFF)_{16}=(1023,96875)_{10}$ až $(8000)_{16}=(-1024)_{10}$
Q4	$2^4=16$	$(7FFF)_{16}=(2047,9375)_{10}$ až $(8000)_{16}=(-2048)_{10}$
Q3	$2^3=8$	$(7FFF)_{16}=(4095,875)_{10}$ až $(8000)_{16}=(-4096)_{10}$
Q2	$2^2=4$	$(7FFF)_{16}=(8191,75)_{10}$ až $(8000)_{16}=(-8192)_{10}$
Q1	$2^1=2$	$(7FFF)_{16}=(16383,5)_{10}$ až $(8000)_{16}=(-16384)_{10}$
Q0	$2^0=1$	$(7FFF)_{16}=(32767)_{10}$ až $(8000)_{16}=(-32768)_{10}$

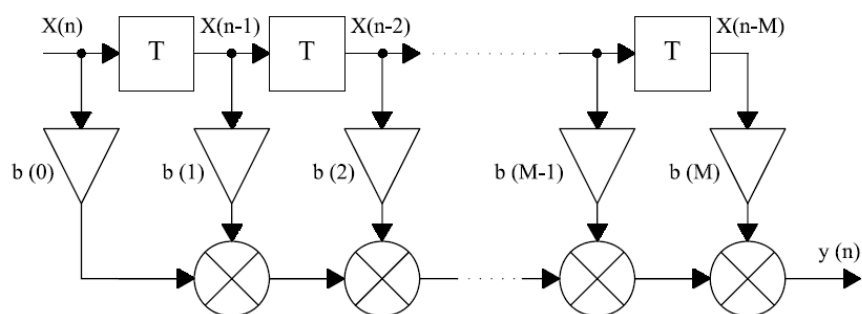
Tab.4.1: Tabulka s váhovými koeficienty pro 16-ti bitové číslo [1]

## 4.2 Implementace FIR filtru

Pokud bude třeba naprogramovat funkci FIR filtru, musí se vycházet z rovnice

$$y[n] = b(0) \cdot X[n] + b(1) \cdot X[n-1] + \dots + b(N-1) \cdot X[n-N+1] \quad (4.5)$$

Pro lepší představu je tato rovnice znázorněna i graficky:



Obr. 4.2: Blokové schéma FIR filtru

Na obrázku lze vidět, jak probíhá výpočet FIR filtru. V horní části obrázku jsou zobrazeny vstupní vzorky  $x[n]$ , které se postupně obnovují s každým novým vzorkem. To znamená, že aktuální vstupní vzorek  $x[n]$  je násobený koeficientem  $b(0)$ . Jestliže přijde nový vstupní vzorek, bude stávající vzorek posunutý o jedno zpoždění doprava. Nový vzorek bude násobený koeficientem  $b(0)$  a vzorek, jenž byl dříve posunutý o zpoždění, bude násobený koeficientem  $b(1)$ . Veškeré načtené vzorky jsou násobeny koeficienty. Vynásobené vzorky se sečtou do jedné hodnoty, která je výstupem filtru. Výstup filtru je označen jako  $y[n]$ . Po provedení násobení a sečtení jednotlivých vzorků jsou vstupní vzorky posunuty vždy o jedno zpoždění.

Před aplikováním filtru je nutné vymezit jak paměťové místo v programové paměti pro koeficienty impulsové odezvy o hodnotě  $M$ , tak paměťové místo pro uložení  $n-M$  vstupních vzorků v datové paměti.

Aplikuje-li se tedy FIR filtr, je nutné mít v datové paměti vymezené registry pro ukládání vstupních vzorků od A/D jednotky. Takové paměťové místo se nazývá buffer a jeho délka je stejná jako řád filtru. Při zpracování algoritmu jsou všeobecně používané dva typy bufferu:

- Lineární buffer,
- Cirkulární buffer.

### Lineární buffer

V datové paměti je vymezená část, do které se ukládají vzorky od A/D převodníku. Nové vstupní vzorky jsou ukládány vždy na nejnižší paměťovou adresu a předchozí vzorky jsou posouvány vždy o jedno adresní místo nahoru. Jakmile vstupní vzorek dosáhne nejvyšší nastavené adresy, dojde k jeho smazání.

### Cirkulární buffer

U cirkulárního bufferu, také nazývaném jako kruhový buffer, je nejstarší vzorek v paměti přepsaný novým vstupním vzorkem. Takto nevzniká posun vzorku, ale mění se jen adresa pro ukládání od nejnižší adresy po nejvyšší.

## 4.2.1 Implementace FIR filtru v jazyce Assembler

Stále velmi oblíbeným programovacím jazykem pro mikropočítače a mikroprocesory je Assembler. Obliba jazyku Assembler je dána především jeho dobrou a přehlednou strukturou, která přímo určuje, jak se daný program bude chovat.

Pro vysvětlení algoritmu FIR filtru postačí jednoduchý příklad v jazyce Assembler. Nejprve je nutné vytvořit a definovat pole proměnných pro vstupní vzorky. Dále se pokračuje definicí řádu daného filtru a na závěr bude definováno pole koeficientů.

```
TAPS .set 4                ; FIR – řád filtru +1
xn .usect "D_LINE",TAPS    ; pole vzorků v 11Q15
.data                     ; FIR – koeficienty v 11Q15
```

Nyní je potřeba vložit do datového pole koeficienty, což se provede tak, že z vypočtených koeficientů se udělá formát čísla Q15.

```
tbl      .word 32768*707/1000      ; 0.707
        .word 32768*123/1000      ; 0.123
        .word 32768*(-175)/1000   ; -0.175
        .word 32768*345/1000      ; 0.345
```

Následuje samotný výpočtový algoritmus. Lze vidět, že na adrese XAR7 jsou uloženy koeficienty a na adrese XAR6 je cyklický buffer s uloženými vzorky. Pro výpočet filtru se může použít instrukce MAC nebo v tomto případě DMAC.

```
DMAC      ACC:P,*XAR6%++,*XAR7++
```

Tento příkaz násobí dva koeficienty v XAR7 se dvěma vzorky XAR6 za jeden strojový cyklus a ukládá je do proměnné P. Příkazem

```
ADDL      ACC:P
```

provádíme jejich součet, který je poslán na výstup D/A převodníku. Celý program je možné vidět na následujících řádcích

```
TAPS .set 4                      ; FIR – řád filtru +1
      xn .usect "D_LINE",TAPS    ; pole vzorků v I1Q15
      .data                      ; FIR – koeficienty v I1Q15
tbl   .word 32768*707/1000      ; 0.707
      .word 32768*123/1000      ; 0.123
      .word 32768*(-175)/1000   ; -0.175
      .word 32768*345/1000      ; 0.345

.text

FIR:    SETC SXM
CLRC    OVM
SPM      1
MOVL    XAR7,#tbl              ; ukazatel na adresu koeficientů
MOVL    XAR6,#xn                ; ukazuje, kde je cirkulární buffer
MOV     AR1,#TAPS-1             ; buffer offset
MOV     *XAR6%++,*(0:adc)       ; definuje místo uložení nového vzorku (x(n))
ZAP     A                      ; maže ACC,P,OVC
RPT     #(TAPS/2)-1             ; RPT další instr.(#+1)time
||DMAC ACC:P,*XAR6%++,*XAR7++ ; výpočet koeficientů se vzorky
ADDL    ACC:P                  ; sečte jednotlivé násobky
MOV     *(0:dac),AH            ; odesílá výslednou hodnotu (y(n))
RET
```

Jak již bylo řečeno, tato diplomová práce se zaměřuje na signálový procesor TMS320F2812 od firmy Texas Instrument. Pro výpočet filtru poskytuje firma pro procesor přímo knihovnu v Assembleru. Knihovna, jejíž název je FIR 16, je velmi optimalizována a její použití je velmi jednoduché. Lze ji také kombinovat s programovacím jazykem C. Nejdůležitější vlastností knihovny je to, že maximální možný řád filtru je 512 koeficientů. Jelikož jsou koeficienty uloženy v 16-ti bitovém čísle a registry a ALU jednotka je 32-bitová, jsou vždy dvě čísla spojená dohromady. Prvních 16 bitů značí jedno uložené číslo a druhá polovina bitů druhé číslo (tento způsob byl již popsán v kapitole 1.1.1). Díky tomu je potřeba o polovinu menší paměť, a to jak pro koeficienty, tak pro vzorky. Navíc se při jednom cyklu vypočtou dva vzorky se dvěma koeficienty, tudíž je množství operací při výpočtu také o polovinu méně a proto procesor vypočte daný filtr rychleji. Fungování knihovny již bylo znázorněno.

Nyní budou funkce definovány v hlavním programu. Ze všeho nejdřív je ale nutné vytvořit paměťový prostor pro cyklický buffer a vymezit sekci pro koeficienty. To se provede pomocí definice

```
firldb      align(0x100) > DRAMH0 PAGE = 1 // buffer (256*2)
firfilt     :>DRAMH0      PAGE = 1 // paměťový prostor koeficientů
```

v knihovně s příponou cmd. Paměťový prostor koeficientů se musí zarovnat na 256 buněk. Pak se v hlavním programu definuje délka filtru. Zde je pro  $N=50$

```
#define FIR_ORDER 50.
```

Následně bude nadefinováno a vloženo jednorozměrné pole s koeficienty

```
#define FIR16_COEFF {k0, k1,-20, k2,-19, kmN, -(N-1)/2-(mN-1)}
```

Tyto 16-ti bitové koeficienty jsou párovány do 32-bitového čísla. Dále budou pojmenovány jednotlivé paměťové místa s koeficienty a vzorky

```
#pragma DATA_SECTION(lpf, "firfilt");
FIR16 lpf = FIR16_DEFAULTS;
#pragma DATA_SECTION(dbuffer, "firldb");
long dbuffer[(FIR_ORDER+2)/2];
```

Zároveň je nutné do hlavního programu zadat příkazy k nastavení procesoru.

```
main()
{
    lpf.dbuffer_ptr=dbuffer;
    lpf.coeff_ptr=(long *)coeff;
    lpf.order=FIR_ORDER;
    lpf.init(&lpf);
}
```

a nakonec v přerušovacím programu bude vyvolána funkce pro výpočet

```
void interrupt fir // přerušovací rutina
```

```
{
```

```
    lpf.input=xn; //načtení vstupní hodnoty  $x[n]$ 
```

```
    lpf.calc(&lpf); // výpočet filtru
```

```
    yn=lpf.output; //vložení výstupní hodnoty na proměnnou  $y[n]$ 
```

```
}
```

kde se nejprve načte vstupní vzorek  $x[n]$  a pak se zavolá funkce `.calc` nacházející se v knihovně FIR16. Ta provede výpočet. Na závěr se vypíše výstupní hodnota  $y[n]$ .

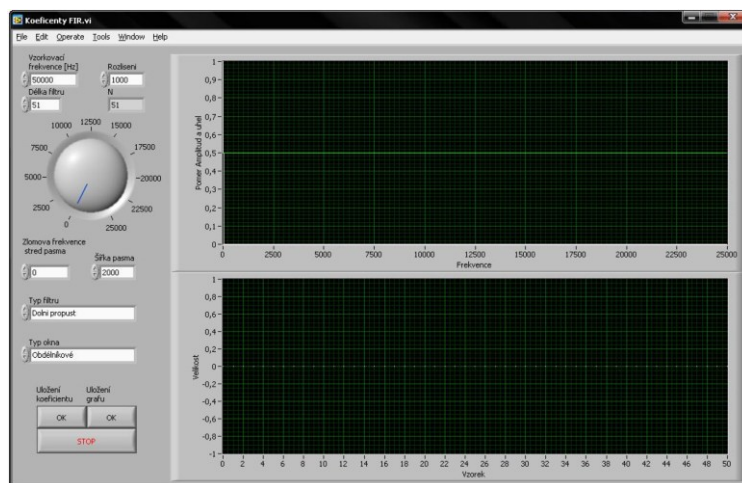
## 4.3 Výpočet koeficientu pro FIR filtr

Použije-li se FIR filtr, je jeho délka pro větší strmost amplitudové charakteristiky o velikosti několika desítek řádů. Proto je pro výpočet koeficientů vhodné využít některý výpočetní program. Z internetových stránek firmy Texas instrument je možné si stáhnout výpočet koeficientů v prostředí Matlab. Tento výpočetní algoritmus je přímo určen k výpočtu koeficientů pro daný procesor. Jednoduchým ovládáním lze nastavit požadovaný druh filtru, zlomové frekvence, délku filtru či jakým oknem bude násoben.

Pro výpočty koeficientu FIR filtru se používají funkce z Matlabu. Získaná čísla jsou následně převedená do datového typu Q15 a sloučená na 32 bitové číslo. Nakonec jsou koeficienty uloženy do textového souboru, kde stačí tyto koeficienty jen zkopírovat a vložit do hlavního programu.

### 4.3.1 Program pro výpočet koeficientu pro FIR filtr

S cílem zjednodušit výpočet koeficientu filtru byl v prostředí LabVIEW vytvořen jednoduchý program, který umí převést koeficienty FIR filtru na koeficienty používané knihovnou FIR16. Podobu tohoto programu znázorňuje obrázek 4.3.

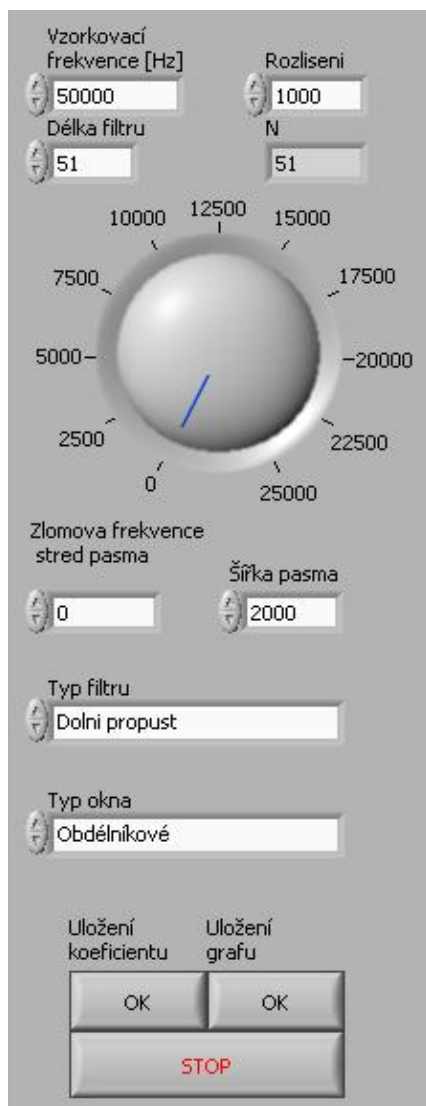


Obr. 4.3: Program pro výpočet koeficientů FIR



V levé části programu jsou nastavovací prvky a v pravé části jsou pod sebou umístěny dva grafy. Vrchní graf ukazuje přenosové pásmo a spodní graf zobrazuje vypočtenou impulzní odezvu.

#### 4.3.1.1 Popis hlavního panelu programu



**Obr. 4.4: Hlavní panel programu**

##### Vzorkovací frekvence

Zde lze nastavit vzorkovací frekvenci A/D převodníku v Hz. V teoretické části je tato frekvence označovaná jako  $f_v$ .

##### Rozlišení

Rozlišení udává kolik bodů bude vykresleno v horním grafu

##### Délka filtru

Řád filtru je možné nastavit od 1 až po 511 s tím, že program vždy při vložení sudého řádu přičte jedničku, aby byl řád lichý.

## Zlomová frekvence

Je zde určována zlomová frekvence filtru a jde nastavit až na polovinu vzorkovací frekvence v Hz.

## Šířka pásma

Šířka pásma se nastavuje tehdy, když používáme typ filtru pásmovou zadrž nebo propust. Nejprve se zadá zlomová frekvence, která bude střed pásma a následně se určí šířka pásma vše v Hz.

## Typ filtru

Nabízí se tu čtyři možnosti výběru:

- dolní propust,
- horní propust,
- pásmová zadrž,
- pásmová propust.

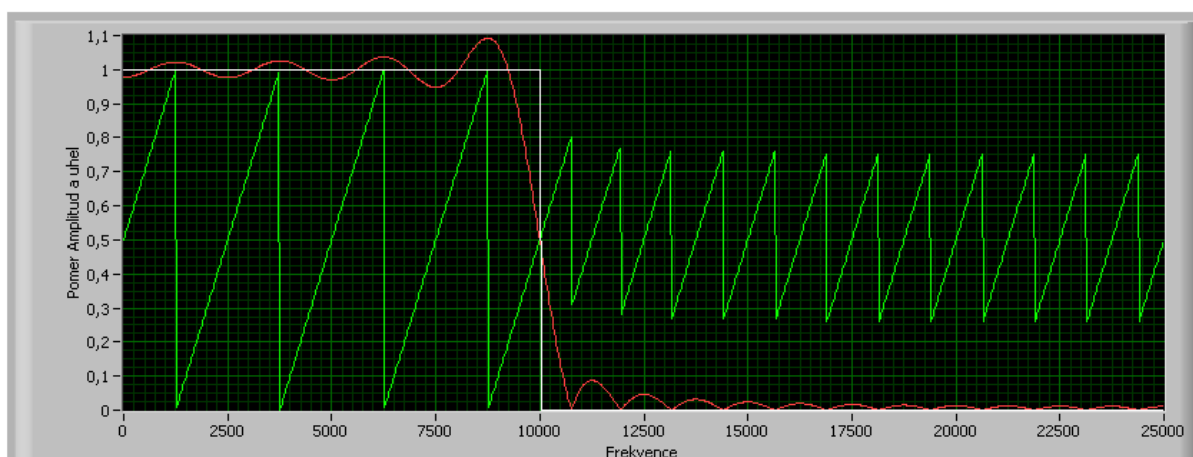
## Typ okna

Typ okna umožňuje nastavit, jakým oknem budou výsledné vzorky násobeny.

- Obdélníkové okno,
- Bartletovo okno,
- Hannovo okno,
- Hammingovo okno,
- Blackmanovo okno.

K uložení koeficientů se používá tlačítko Ok pod nadpisem Uložení koeficientu. Vyskočí okno, kde se určí místo a název uložení vypočtených koeficientů. Takovýto soubor se ukládá ve formátu txt. Nachází se zde také tlačítko pro uložení hodnot z vrchního grafu ukládající data v txt formátu. Pro opuštění programu pak stačí stisknout křížek v pravém horním rohu aplikace.

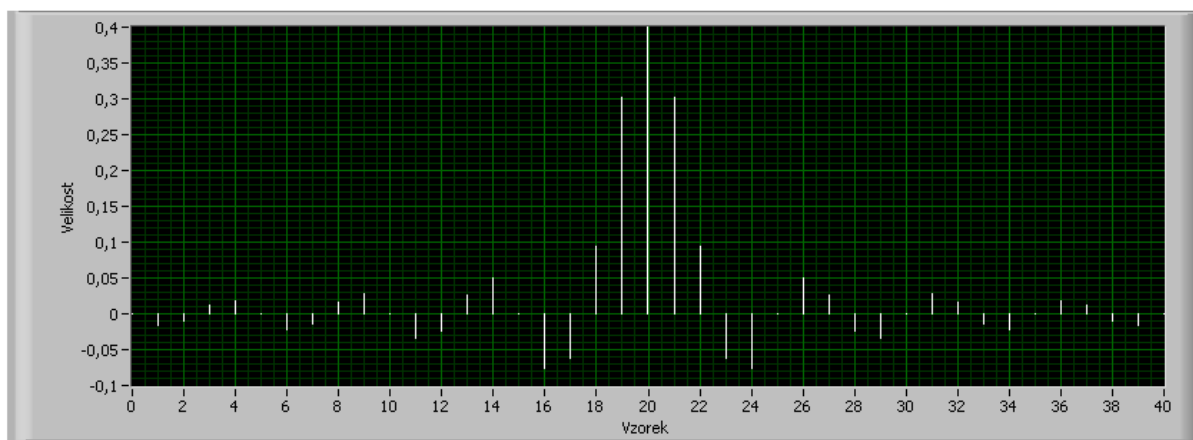
## Zobrazovací část:



**Obr. 4.4: Zobrazovací část frekvenční charakteristiky**

Graf zobrazuje frekvenční charakteristiku, kde na ose x je vyznačená frekvence  $f_v$  [Hz] a na ose y zesílení A. Křivky v grafu mají bílou, červenou a zelenou barvu, z nichž každá barva značí něco

jiného. Bílá barva představuje ideální frekvenční charakteristiku dolní propusti. Červená barva označuje skutečnou frekvenční charakteristiku daného filtru. Zelená barva je fázová kmitočtová charakteristika

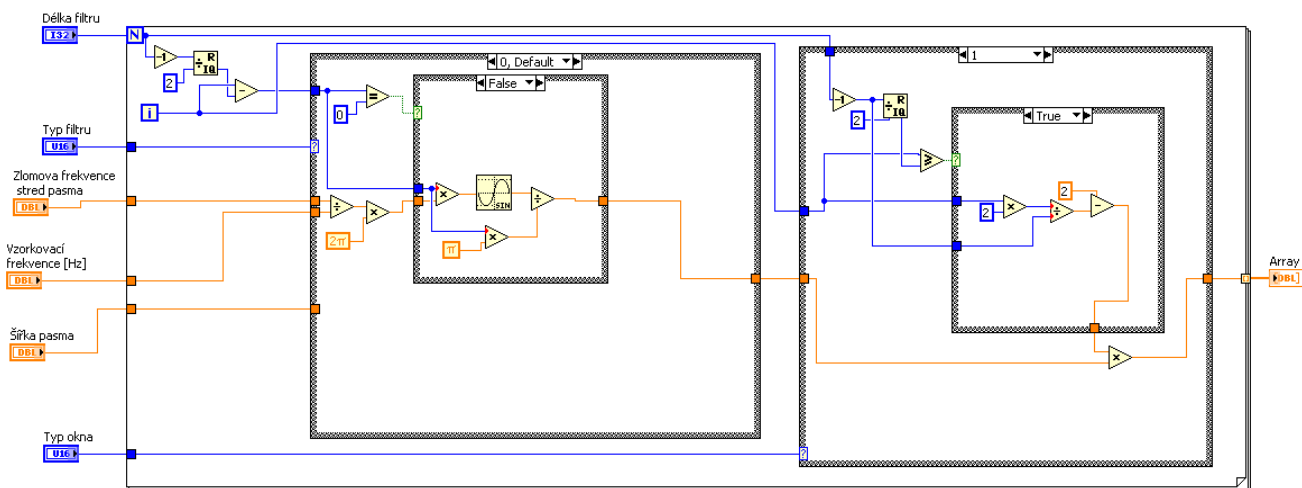


**Obr. 4.4: Zobrazovací část koeficientů**

Uvedený graf znázorňuje symetrickou lichou impulzovou odezvu pro daný filtr. Na ose  $x$  je vyznačeno pořadí koeficientů a na ose  $y$  jsou pak uvedeny hodnoty koeficientů.

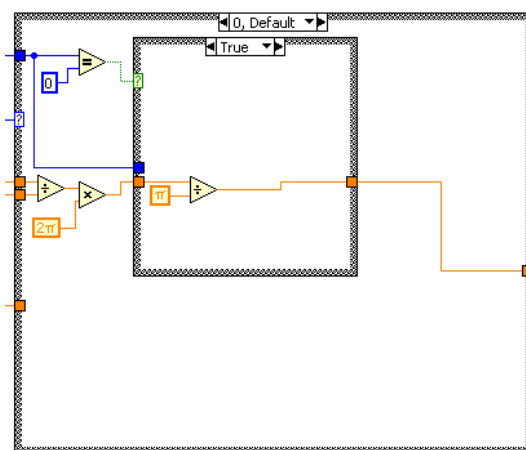
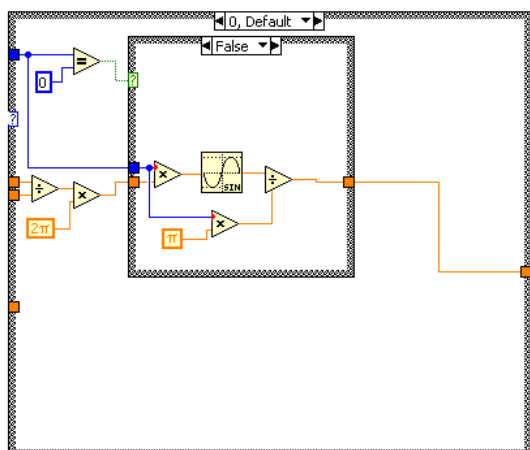
#### 4.3.1.2 Popis programové struktury vytvořené v LabView

Popis metody a odvození matematických vztahů jsou popsány v kapitole 3.4.2. Proto zde bude ukázáno pouze to, jak jsou jednotlivé filtry navrhнуты v prostředí LabView.

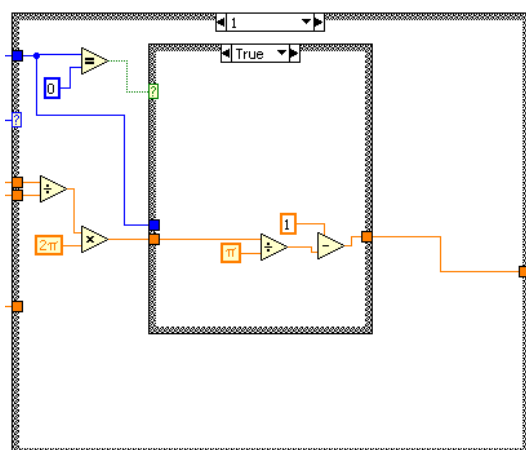
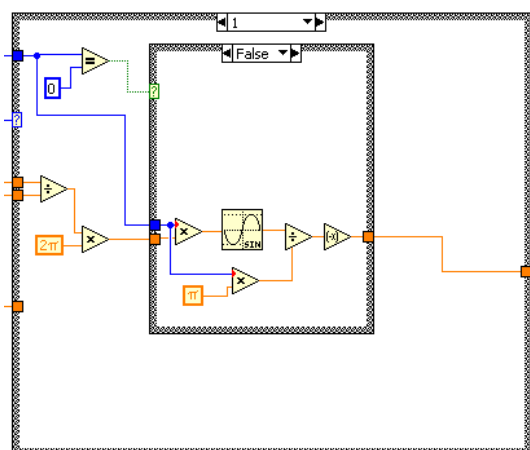


**Obr 4.5: Celkový pohled na výpočtový algoritmus koeficientů**

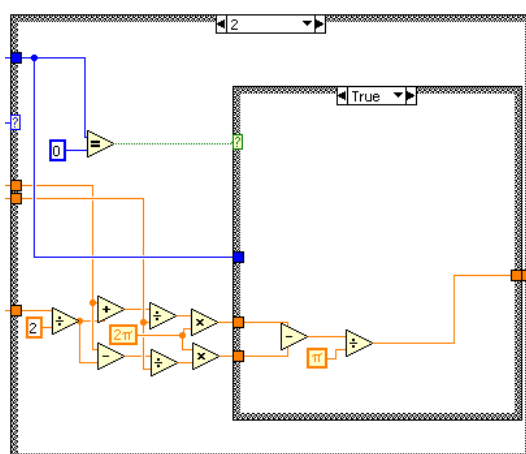
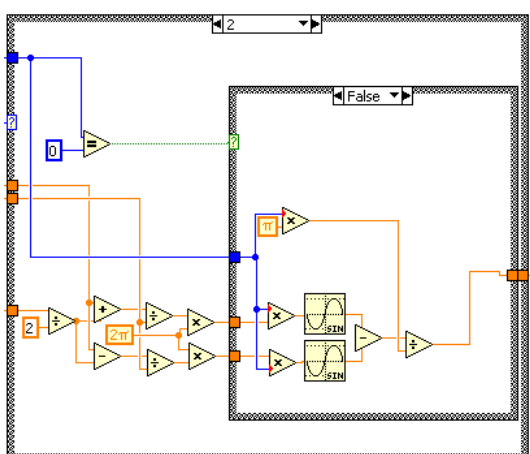
Na obrázku 4.5 jsou dvě case struktury. Struktura na levé straně obrázku určuje, jaký typ filtru bude používán. Tato case struktura v sobě obsahuje i druhou case strukturu, která zajišťuje správné výpočty koeficientů při dělení nulou. Jednotlivé case struktury pro různé typy filtru jsou zobrazeny na následujících obrázcích.



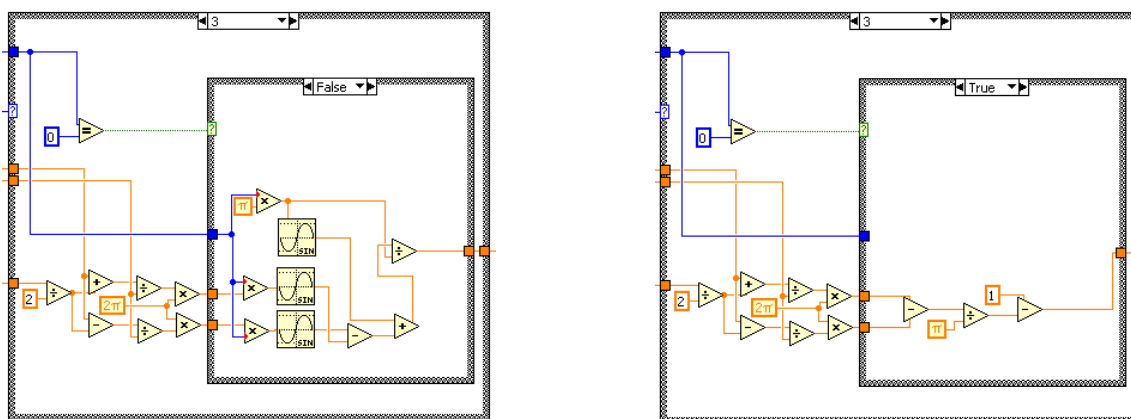
*Obr. 4.6: Dolní propust*



*Obr. 4.7: Horní propust*

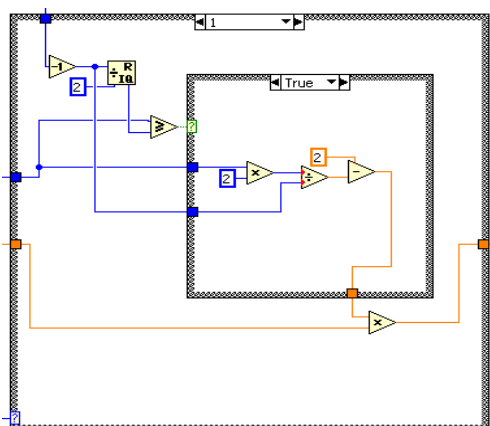


*Obr. 4.8: Pásmová propust*

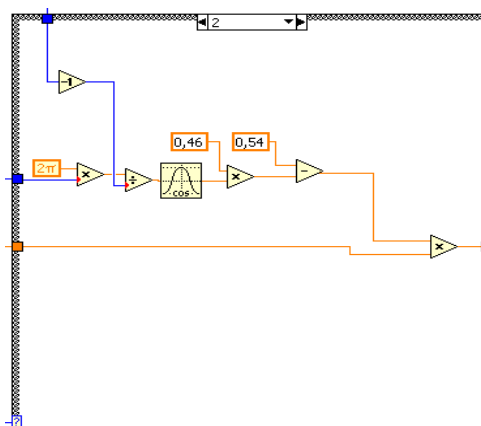


**Obr. 4.9: Pásmová zádrž**

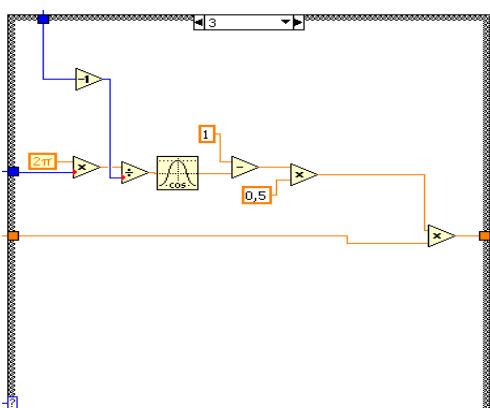
Po výpočtu koeficientů se uplatní druhá case struktura. Ta určuje, jaký druh okna bude pro výpočet koeficientů použitý. Obrázky uvedené níže ukazují podobu jednotlivých oken.



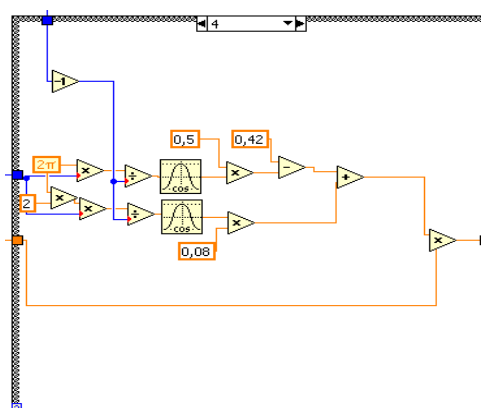
**Obr. 4.10: Bartletovo okno**



**Obr. 4.11: Hammingovo okno**



**Obr. 4.12: Hannovo okno**



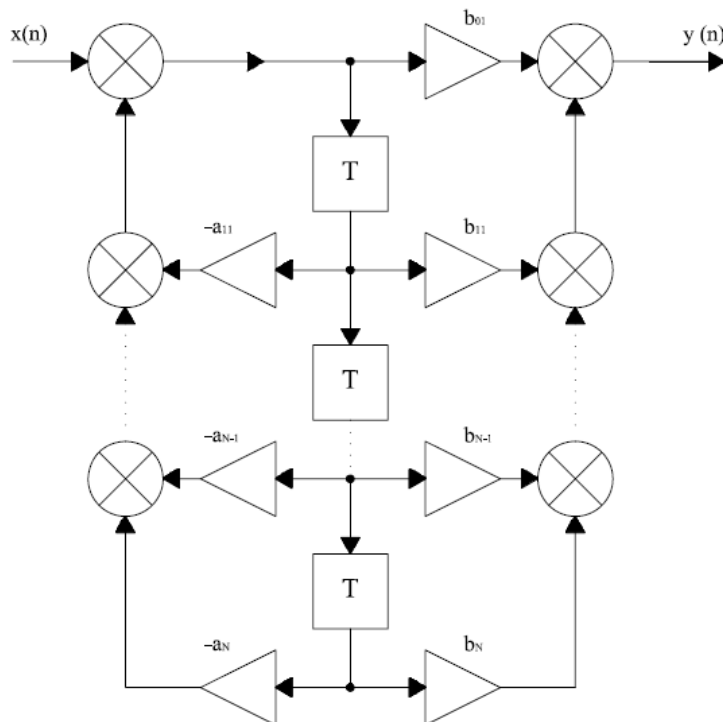
**Obr. 4.13: Blackmanovo okno**

## 4.4 Implementace IIR filtru

Pro návrh filtru typu IIR se vychází z diferenční rovnice

$$y[n] = \sum_{i=0}^M b_i x[n-i] - \sum_{i=1}^N a_i x[n-i]. \quad (4.6)$$

Bloková struktura rovnice se nazývá přímá kanonická forma a je vyobrazena na obrázku 4.14.



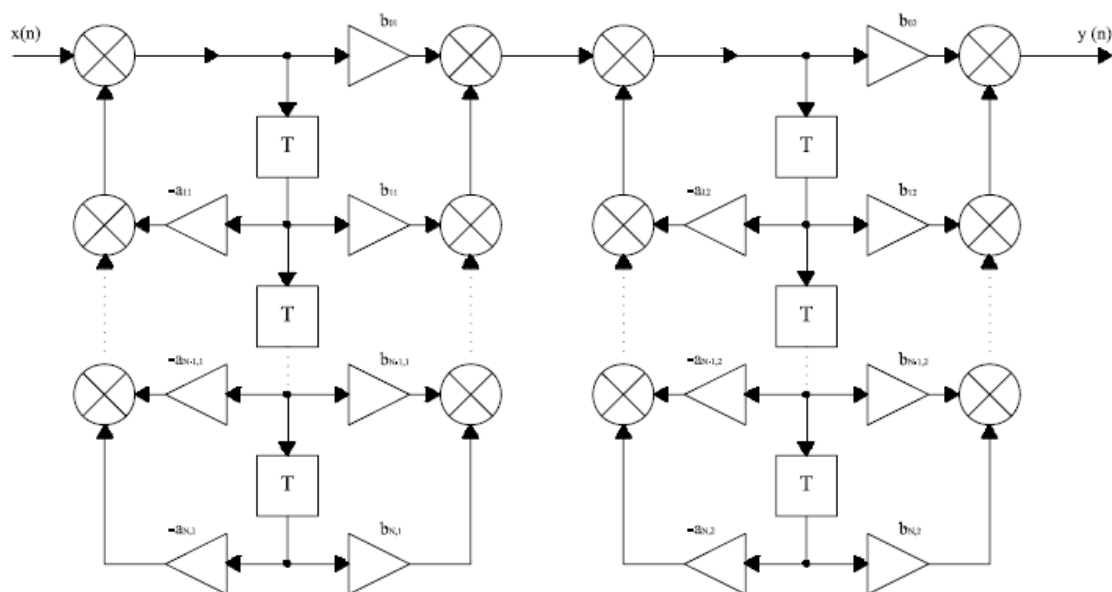
**Obr.4.14: Blokové schéma přímé kanonické formy  $N$ -tého řádu**

Pro lepší názornost je uvedena přenosová funkce obecného filtru přímé kanonické formy:

$$H[z] = \frac{b_0 + b_1 z^{-1} + \dots + b_N z^{-(N-1)} + b_N z^{-N}}{1 + a_1 z^{-1} + \dots + a_N z^{-(N-1)} + a_N z^{-N}} \quad (4.7)$$

Jak je vidět z blokového schématu, řád filtru je tvořen jedním zpožďovacím členem a dvěma násobičkami. Následkem takové struktury je zvýšený počet aritmetických operací a tudíž i požadavek na výpočtový výkon bude větší. Na rozdíl od filtrů FIR, obsahuje IIR filtr menší počet řádů a proto zde je úspora paměťového místa.

K omezení velikosti řádů filtru se používají dvě sériově spojené přímé kanonické formy. Toto spojení se označuje jako druhá kanonická forma, která je zobrazena níže.



**Obr. 4.15: Blokové schéma druhé kanonické struktury**

### Implementace v jazyce Assembler

Jak už bylo napsáno výše, při výpočtu IIR filtru je třeba provést více aritmetických operací než je tomu u filtru typu FIR. K výpočtu jsou použity upravené rovnice (4.8 a 4.9), které vycházejí z rovnice (4.6)

$$d[n] = x[n] - a_1 d[n-1] - a_2 d[n-2] \quad (4.8)$$

$$y[n] = b_0 d[n] + b_1 d[n-1] + b_2 d[n-2] \quad (4.9)$$

kde nejprve program vypočte první a pak druhou rovnici.

I v tomto případě firma Texas Instrument dodává již hotový program pro IIR filtr. Samotný výpočet je opět napsán v jazyku Assembler. Program je implementován jako druhá kanonická struktura s jednotlivými sekcemi druhého řádu. Pro IIR filtr jsou vytvořeny knihovny se jmény IIR16 a IIR32. První knihovny používá typ čísel INT, což je 16-ti bitové číslo. U druhé zmíněné knihovny je použitý typ čísel long, tedy 32-bitové číslo. Tato diplomová práce se zaměřuje na knihovnu IIR16 a proto zde budou popsány její základní funkce.

Před samotným popisem základních funkcí knihovny IIR 16 je třeba definovat počáteční podmínky filtru a určení, kde budou ukládány jednotlivé vzorky. Toto zajišťuje funkce `_IIR5BIQ16_init`, která se spustí při inicializaci procesoru.

`_IIR5BIQ16_init:`

```

ADDB      XAR4,#          ; XAR4->dbuffer_ptr
MOVL      XAR6,*+XAR4[0]   ; XAR6=dbuffer_ptr
MOV        ACC,*+XAR4[2]<<1 ; ACC=2*nbiq
SUB        ACC,#1          ; ACC=(2*nbiq)-1
MOVZ      AR0,AL          ; AR0=(2*nbiq)-1

```

```

ADDL    ACC,*XAR4      ; ACC=dbuffer_ptr + (2*nbiq) - 1
MOVL    *XAR4,ACC      ; XAR4->dbuffer_ptr,
                        ; dbuffer_ptr=dbuffer_ptr + (2*nbiq) - 1

MOV      *+XAR4[3],#0   ; input=0
MOVB     ACC,#16
SUB      ACC,*+XAR4[5]
MOV      *+XAR4[5],AL   ; qfmat=16-qfmat
MOV      *+XAR4[6],#0   ; output=0
RPT      AR0
|| MOV    *XAR6++,#0
LRETR

```

Pro samotný výpočet filtru je zde vyvolána funkce *\_IIR5BIQ16\_calc*. Na první pohled jde vidět, že obsahuje více příkazů než FIR filtr. Nejprve se načtou jednotlivé koeficienty na adresu XAR7 a zavede se cyklický buffer na adresu XAR6. Vstupní vzorky jsou posílány na adresu AR0.

*\_IIR5BIQ16\_calc*:

```

SETC     SXM,OVM
ZAPA                                           ; Zero the ACC, P registers and OVC counter
ADDB     XAR4,#4                             ; XAR4->coeff_ptr
MOVL     XAR7,*XAR4++                        ; XAR4->dbuffer_ptr, XAR7=coeff_ptr->a21
MOVL     XAR6,*XAR4++                        ; XAR4->order, XAR6=dbuffer_ptr->d1(n-2)
MOVZ     AR0,*XAR4++                         ; XAR4->input, AR0=nbiq
MOV      T,*XAR4++                          ; XAR4->isf, T=input
MPY      ACC,T,*XAR4++                      ; XAR4->qfmat, ACC=input*isf
ADDB     XAR6,#1
SUBB     XAR0,#1                            ; AR0=nbiq-1

```

pak jsou už v kódu aplikované rovnice (4.8) a (4.9), které jsou upravené na tvar

$$dk[n] = x[n] - a_{1k} dk[n-1] - a_{2k} dk[n-2], \quad (4.10)$$

$$y[n] = b_{0k} dk[n] + b_{1k} dk[n-1] + b_{2k} dk[n-2] \quad (4.11)$$

; kth Biquad Computation, where k=1:nbiq biqd

```

MOV      T,*--XAR6      ; T=dk(n-2)
MPY      P,T,*XAR7++    ; P= dk(n-2)*a2k
MOV      T,*--XAR6      ; T=dk(n-1)
MPYA     P,T,*XAR7++    ; ACC=input*isf + dk(n-2)*a2k
                        ; P=dk(n-1)*a1k

MOV      T,*+XAR6[1]    ; T=dk(n-2)
MPYA     P,T,*XAR7++    ; ACC=input*isf + dk(n-2)*a2k + dk(n-1)*a1k
                        ; P=dk(n-2)*b2k

MOV      T,*XAR4        ; T=qfmat
LSLL     ACC,T          ; ACC=input*isf + dk(n-2)*a2k + dk(n-1)*a1k (Q15)
MOVZ     AR5,AH         ; AR5=dk(n)=input*isf + dk(n-2)*a2k + dk(n-1)*a1k
(Q15)    MOV      ACC,#0 ; ACC=0

```



```

MOVAD    T,*XAR6           ; T=dk(n-1), dk(n-2)=dk(n-1), ACC=dk(n-2)*b2k
MPY      P,T,*XAR7++       ; P=dk(n-1)*b1k,
MOV      T,AR5             ; T=dk(n)
MPYA     P,T,*XAR7++       ; ACC=dk(n-1)*b1k + dk(n-2)*b2k, P=dk(n)*b0k
ADDL     ACC,P
MOV      *XAR6,T           ; dk(n-1)=dk(n)
BANZ     biqd,AR0--
MOV      T,*XAR4++         ; T=qfmat, XAR4->output
LSLL     ACC,T
ROR      ACC
MOV      *XAR4,AH          ; output=Filtered Output in Q14 format
CLRC     OVM
LRETR

```

## Implementace

IIR filtr se v hlavním programu nastavuje podobně jako FIR filtr. Opět se nejdříve vymezí paměťová sekce pro data v souboru s příponou cmd.

```
iirfilt      :> DRAMH0      PAGE = 1
```

Opět jsou vloženy koeficienty, ale tentokrát převedené na 16-ti bitové číslo.

```

#define IIR16_COEFF {\
    -6197,14244,9,18,9,\
    -6469,14513,37,74,37,\
    -6999,15038,25,51,25,\
    -7752,15784,13420,26839,13420}

```

K výpočtu koeficientů. jsou nadefinovány další parametry.

```

#define IIR16_ISF 145      // měřítkový koeficient zajišťuje nepřekročení rozsahu signálu
#define IIR16_NBIQ 4       // počet sekcí druhého řádu
#define IIR16_QFMAT 13     // Q formát vložených koeficientů (Q13)

```

a dále jsou nadefinovány jednotlivé paměťové prostory

```

#pragma DATA_SECTION(iir, "iirfilt");
IIR5BIQ16 iir=IIR5BIQ16_DEFAULTS; // načte výchozí vstupní data

#pragma DATA_SECTION(dbuffer, "iirfilt");
int dbuffer[2*IIR16_NBIQ]; /*velikost bufferu , každá sekce má dvě zpoždovací hodnoty*/
/* definice pole konstant a místo uložení v paměti
const int coeff[5*IIR16_NBIQ]=IIR16_LPF_COEFF;

main()

```

```

{
    iir.dbuffer_ptr=dbuffer; /* Object Initialization */
    iir.coeff_ptr=(int *)coeff;
    iir.qfmat=IIR16_QFMAT;
    iir.nbiq=IIR16_NBIQ;
    iir.isf=IIR16_ISF;
    iir.init(&iir);
}

```

Jak u filtru typu FIR, tak u IIR filtru je napsána stejná přerušovací rutina.

```

void interrupt iir // přerušovací rutina IIR
{
    iir.input=xn; // načtení vstupní hodnoty x[n]
    iir.calc(&iir); // výpočet filtru
    yn=iir.output; // vložení výstupní hodnoty na proměnnou y[n]
}

```

## 5 MĚŘENÍ

V této kapitole bude na vzorových příkladech znázorněn výpočet jednotlivých koeficientů FIR filtru a IIR filtru. Výpočty budou následně ověřeny pomocí měření.

### 5.1 Příklad výpočtu koeficientů FIR filtru

K výpočtu koeficientů bude použita dolní propust s parametry vzorkovací frekvence  $f_v = 50\text{kHz}$ , zlomová frekvence  $f_d = 2\text{kHz}$  a řád filtru  $N=41$ . Koeficienty budou vypočteny pomocí metody Fourierových řad a vynásobeny okny.

Výpočet úhlového kmitočtu diskretního signálu:

$$\omega_d = 2\pi \frac{f_d}{f_v} = 2\pi \frac{2000}{50000} = \underline{0,251 \text{ rad} \cdot \text{s}^{-1}}, \quad (5.1)$$

$$m = n - \frac{N-1}{2} = 14 - \frac{41-1}{2} = \underline{-6}. \quad (5.2)$$

**VÝPOČET KOEFICIENTU:**

Pro  $m=0$

$$h_d[20] = \frac{\omega_d}{\pi} = \frac{0,251}{\pi} = \underline{0,08}. \quad (5.3)$$

Pro  $m \neq 0$

$$h_d[14] = \frac{\sin(\omega_d m)}{m\pi} = \frac{\sin(0,251 \cdot (-7))}{-7 \cdot \pi} = \underline{0,05295}. \quad (5.4)$$

**POUŽITÁ OKNA:**

▪ **Obdélníkové okno:**

Pro  $0 \leq n \leq N-1$ :

$$h_w[14] = h_d[14] \cdot w[14] = 0,0529 \cdot 1 = \underline{0,0529}. \quad (5.5)$$

▪ **Bartletovo okno:**

Pro  $0 \leq n \leq (N-1)/2$ :

$$h_w[n] = h_d[n] \cdot w[n] = h_d[n] \cdot \frac{2n}{N-1} = 0,0529 \cdot \frac{2 \cdot 14}{41-1} = \underline{0,0371}. \quad (5.6)$$

▪ **Hannovo okno:**

Pro  $0 \leq n \leq N-1$ :

$$h_w[n] = h_d[n] \cdot w[n] = h_d[n] \cdot 0,5 \left[ 1 - \cos\left(\frac{2\pi n}{N-1}\right) \right], \quad (5.7),$$

$$h_w[14] = 0,0529 \cdot 0,5 \left[ 1 - \cos\left(\frac{2\pi \cdot 14}{41-1}\right) \right] = \underline{0,042}. \quad (5.8)$$

▪ **Hammingovo okno:**

Pro  $0 \leq n \leq N-1$ :

$$h_w[n] = h_d[n] \cdot w[n] = h_d[n] \cdot 0,54 - 0,46 \cdot \cos\left(\frac{2\pi n}{N-1}\right), \quad (5.9)$$

$$h_w[14] = 0,0529 \cdot 0,54 - 0,46 \cdot \cos\left(\frac{2\pi \cdot 14}{41-1}\right) = \underline{0,0429}. \quad (5.10)$$

▪ **Blackmanovo okno:**

Pro  $0 \leq n \leq N-1$ :

$$h_w[n] = h_d[n] \cdot w[n] = h_d[n] \cdot 0,42 - 0,5 \cdot \cos\left(\frac{2\pi n}{N-1}\right) + 0,08 \cdot \cos\left(\frac{4\pi n}{N-1}\right), \quad (5.11)$$

$$h_w[14] = 0,0529 \cdot 0,42 - 0,5 \cdot \cos\left(\frac{2\pi \cdot 14}{41-1}\right) + 0,08 \cdot \cos\left(\frac{4\pi \cdot 14}{41-1}\right) = \underline{0,0365}. \quad (5.12)$$

číslo koeficientu	Fourierovy koeficienty	Použitá okna				
		obdélníkové	Bartletovo	Hannovo	Hammingo-vo	Blackmano-vo
n	$h_d[n]$	$h_w[n]$	$h_w[n]$	$h_w[n]$	$h_w[n]$	$h_w[n]$
0	-0,0151	-0,0151	0,0000	0,0000	-0,0012	0,0000
1	-0,0167	-0,0167	-0,0008	-0,0001	-0,0014	0,0000
2	-0,0174	-0,0174	-0,0017	-0,0004	-0,0018	-0,0002
3	-0,0169	-0,0169	-0,0025	-0,0009	-0,0022	-0,0004
4	-0,0153	-0,0153	-0,0031	-0,0015	-0,0026	-0,0006
5	-0,0125	-0,0125	-0,0031	-0,0018	-0,0027	-0,0008
6	-0,0084	-0,0084	-0,0025	-0,0017	-0,0023	-0,0008
7	-0,0031	-0,0031	-0,0011	-0,0008	-0,0010	-0,0004
8	0,0033	0,0033	0,0013	0,0011	0,0013	0,0007
9	0,0107	0,0107	0,0048	0,0045	0,0050	0,0028
10	0,0187	0,0187	0,0094	0,0094	0,0101	0,0064
11	0,0273	0,0273	0,0150	0,0158	0,0167	0,0115
12	0,0360	0,0360	0,0216	0,0236	0,0246	0,0184
13	0,0447	0,0447	0,0290	0,0325	0,0334	0,0268
14	<b>0,0529</b>	<b>0,0529</b>	<b>0,0371</b>	<b>0,0420</b>	<b>0,0429</b>	<b>0,0365</b>
15	0,0605	0,0605	0,0454	0,0517	0,0524	0,0468
16	0,0672	0,0672	0,0538	0,0608	0,0613	0,0571
17	0,0726	0,0726	0,0617	0,0687	0,0690	0,0663
18	0,0767	0,0767	0,0690	0,0748	0,0749	0,0736
19	0,0792	0,0792	0,0752	0,0787	0,0787	0,0784
20	0,0800	0,0800	0,0800	0,0800	0,0800	0,0800

číslo koeficientu	Fourierovy koeficienty	Použitá okna				
		obdélníkové	Bartletovo	Hannovo	Hammingo-vo	Blackmano-vo
n	$h_d[n]$	$h_w[n]$	$h_w[n]$	$h_w[n]$	$h_w[n]$	$h_w[n]$
21	0,0792	0,0792	0,0752	0,0787	0,0787	0,0784
22	0,0767	0,0767	0,0690	0,0748	0,0749	0,0736
23	0,0726	0,0726	0,0617	0,0687	0,0690	0,0663
24	0,0672	0,0672	0,0538	0,0608	0,0613	0,0571
25	0,0605	0,0605	0,0454	0,0517	0,0524	0,0468
26	0,0529	0,0529	0,0371	0,0420	0,0429	0,0365
27	0,0447	0,0447	0,0290	0,0325	0,0334	0,0268
28	0,0360	0,0360	0,0216	0,0236	0,0246	0,0184
29	0,0273	0,0273	0,0150	0,0158	0,0167	0,0115
30	0,0187	0,0187	0,0094	0,0094	0,0101	0,0064
31	0,0107	0,0107	0,0048	0,0045	0,0050	0,0028
32	0,0033	0,0033	0,0013	0,0011	0,0013	0,0007
33	-0,0031	-0,0031	-0,0011	-0,0008	-0,0010	-0,0004
34	-0,0084	-0,0084	-0,0025	-0,0017	-0,0023	-0,0008
35	-0,0125	-0,0125	-0,0031	-0,0018	-0,0027	-0,0008
36	-0,0153	-0,0153	-0,0031	-0,0015	-0,0026	-0,0006
37	-0,0169	-0,0169	-0,0025	-0,0009	-0,0022	-0,0004
38	-0,0174	-0,0174	-0,0017	-0,0004	-0,0018	-0,0002
39	-0,0167	-0,0167	-0,0008	-0,0001	-0,0014	0,0000
40	-0,0151	-0,0151	0,0000	0,0000	-0,0012	0,0000

Tab. 5.1: Vypočtené koeficienty

### Převod čísla na formát Q15:

$$h_{14} = h_w[14] \cdot 2^{15} = 0,05295 \cdot 32768 = \underline{1735}, \quad (5.13)$$

$$h_{35} = h_w[35] \cdot 2^{15} = -0,0125 \cdot 32768 = -\underline{410}. \quad (5.14)$$

Převod na hexadecimální číslo:

$$h_{14} = \underline{1735}_{10} = \underline{6C7}_{16}, \quad (5.15)$$

$$h_{35} = -\underline{410}_{10} = -\underline{19A}_{16}, \quad (5.16)$$

$$h_{35} = h_w[n] \cdot 2^{15} = -19A = FFFF - 19A = \underline{FE65}. \quad (5.17)$$

Složení dvou čísel pro paralelní zpracování:

$$k_{15} = h_{14} \mid h_{35} = \underline{6C7FE65}_{16} = \underline{11370085}_{10} \cong \underline{113770087}_{10}. \quad (5.18)$$

	k		k
0	2621	11	40173917
1	-32503262	12	58523757
2	-35911216	13	77397915
3	-37287604	14	96009966
4	-36370278	<b>15</b>	<b>113770087</b>
5	-32897088	16	130088458
6	-26802489	17	144375253
7	-17955400	18	156040647
8	-6617956	19	164691420
9	7144317	20	170065424
10	22872677		

*Tab. 5.2: Vypočtené koeficienty pro obdélníkové okno*

## 5.2 Změřené průběhy filtru FIR

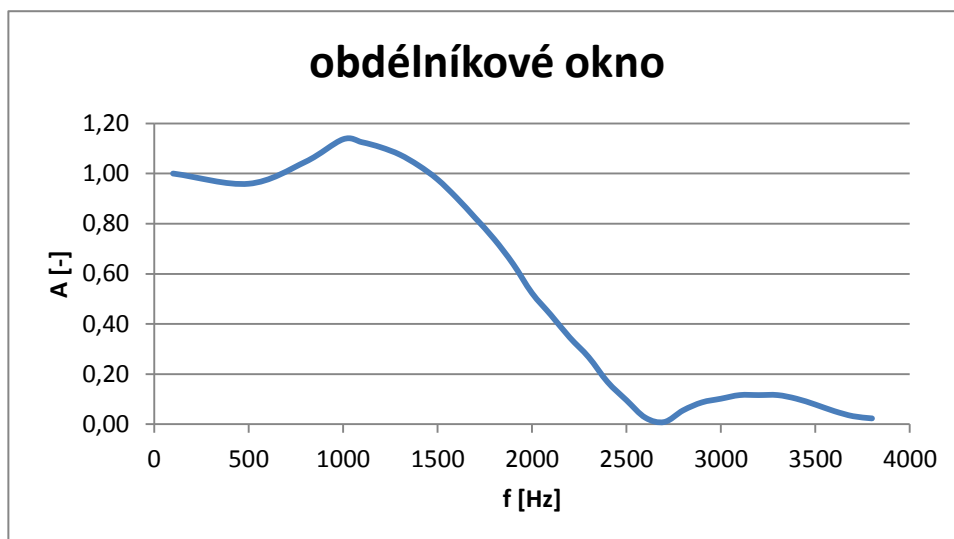
f[Hz]	U <sub>x</sub> [V]	obdélníkové okno		Bartletovo okno		Hammingovo okno	
		U <sub>y</sub> [V]	A	U <sub>y</sub> [V]	A	U <sub>y</sub> [V]	A
100	3,44	3,44	1,00	3,33	0,97	3,40	0,99
500	3,44	3,30	0,96	3,20	0,93	3,33	0,97
800	3,44	3,60	1,05	3,00	0,87	3,20	0,93
1000	3,44	3,91	1,14	2,90	0,84	3,18	0,92
<b>1100</b>	<b>3,44</b>	<b>3,87</b>	<b>1,13</b>	2,82	0,82	3,09	0,90
1200	3,44	3,80	1,10	2,72	0,79	2,99	0,87
1300	3,44	3,70	1,08	2,61	0,76	2,85	0,83
1400	3,44	3,55	1,03	2,50	0,73	2,73	0,79
1500	3,44	3,36	0,98	2,39	0,69	2,60	0,76
1600	3,44	3,11	0,90	2,28	0,66	2,45	0,71
1700	3,44	2,83	0,82	2,15	0,63	2,30	0,67
1800	3,44	2,54	0,74	2,00	0,58	2,13	0,62
1900	3,44	2,20	0,64	1,85	0,54	1,99	0,58
2000	3,44	1,80	0,52	1,70	0,49	1,76	0,51
2100	3,44	1,50	0,44	1,55	0,45	1,64	0,48
2200	3,44	1,19	0,35	1,43	0,42	1,50	0,44
2300	3,44	0,92	0,27	1,28	0,37	1,33	0,39
2400	3,44	0,58	0,17	1,20	0,35	1,14	0,33
2500	3,44	0,33	0,10	0,98	0,28	1,00	0,29
2600	3,44	0,09	0,03	0,86	0,25	0,88	0,26
2700	3,44	0,03	0,01	0,71	0,21	0,74	0,22
2800	3,44	0,19	0,06	0,72	0,21	0,59	0,17
2900	3,44	0,30	0,09	0,62	0,18	0,48	0,14
3000	3,44	0,35	0,10	0,53	0,15	0,38	0,11
3100	3,44	0,40	0,12	0,44	0,13	0,30	0,09
3200	3,44	0,40	0,12	0,37	0,11	0,19	0,06
3300	3,44	0,40	0,12	0,32	0,09	0,13	0,04
3400	3,44	0,35	0,10	0,25	0,07	0,05	0,01
3500	3,44	0,27	0,08	0,22	0,06	0,03	0,01
3600	3,44	0,18	0,05	0,18	0,05	0,01	0,00
3700	3,44	0,11	0,03	0,15	0,04	0,01	0,00
3800	3,44	0,08	0,02	0,13	0,04	0,01	0,00

Tab. 5.3: Naměřené hodnoty FIR filtru

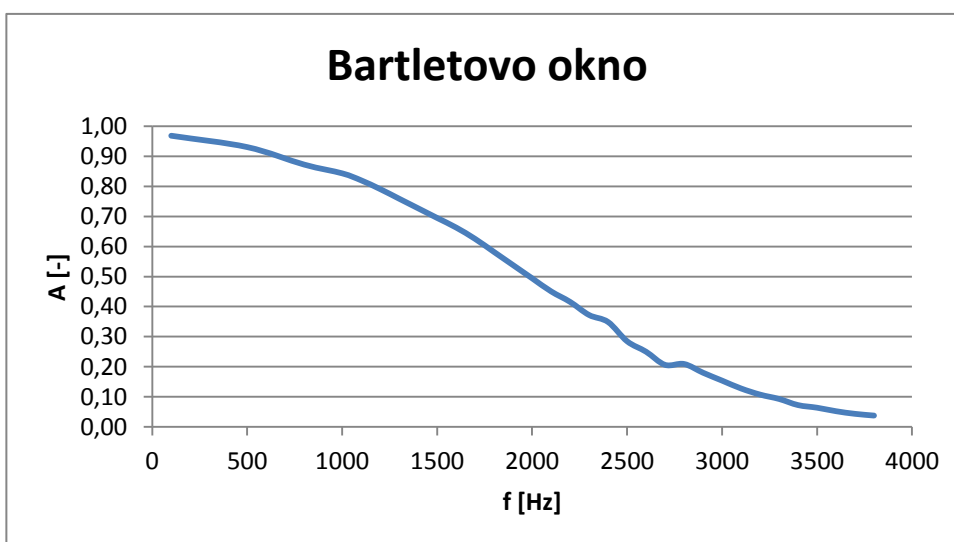
**Příklad výpočtu:**

Zesílení:

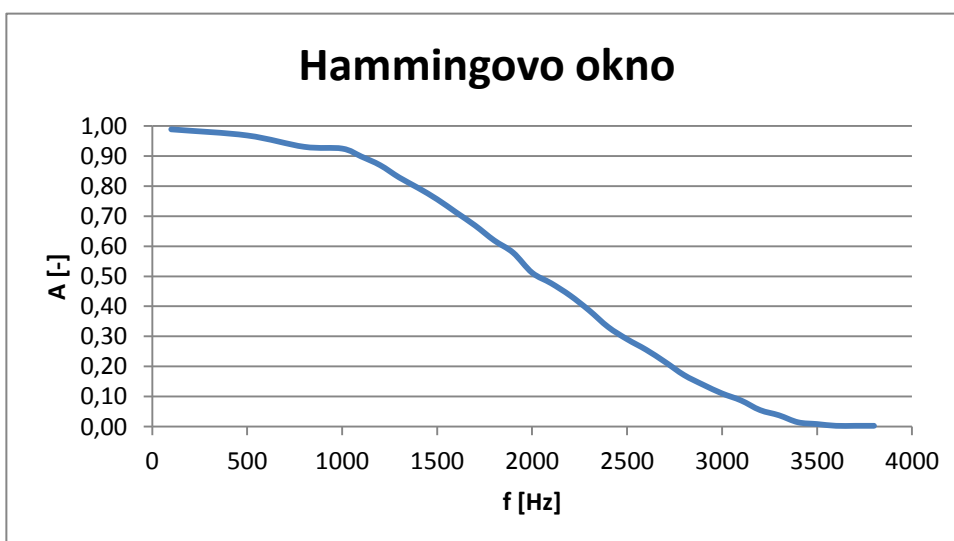
$$A = \frac{U_y}{U_x} = \frac{3,44}{3,8} = \underline{1,1} \quad (5.19)$$



**Obr. 5.1:** Frekvenční charakteristika FIR filtru při použití obdelníkového okna



**Obr. 5.2:** Frekvenční charakteristika FIR filtru při použití Bartletova okna

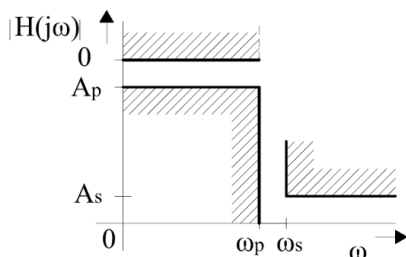


**Obr. 5.3:** Frekvenční charakteristika FIR filtru při použití Hammingova okna



### 5.3 Příklad výpočtu koeficientů IIR filtru

K výpočtu koeficientů IIR bude použita dolní propust s Butterworthovou aproximací. Vzorkovací frekvence je 50kHz s hodnotami tolerančního pásma  $A_p=0,8$ ,  $A_s=0,1$   $f_p=1\text{kHz}$ ,  $f_s=2\text{kHz}$



**Obr. 5.4: Toleranční schéma dolní propusti**

Výpočet úhlového kmitočtu diskretního signálu:

$$\omega_p = 2\pi \frac{f_p}{f_v} = 2\pi \cdot \frac{10^3}{50 \cdot 10^3} = \underline{0,04\pi}, \quad (5.20)$$

$$\omega_s = 2\pi \frac{f_s}{f_v} = 2\pi \cdot \frac{2 \cdot 10^3}{50 \cdot 10^3} = \underline{0,08\pi}. \quad (5.21)$$

Poměr analogových frekvencí:

$$\frac{\omega_s}{\omega_p} = \frac{\text{tg}(\omega_s T / 2)}{\text{tg}(\omega_p T / 2)} = \frac{\text{tg}(0,08 / 2)}{\text{tg}(0,04 / 2)} = \underline{4,236}. \quad (5.22)$$

Výpočet řádů filtru:

$$N \geq \frac{\log \left[ \frac{(1/A_s^2) - 1}{(1/A_p^2) - 1} \right]}{\log \frac{\omega_s}{\omega_p}}, \quad (5.23)$$

$$N \geq \frac{\log \left[ \frac{(1/0,01) - 1}{(1/0,64) - 1} \right]}{\log 4,236} \geq 1,79. \quad (5.24)$$

Pro návrh filtru postačí druhý řád (N=2). Pro názornost bude dále uveden postup, jak vypočíst filtr v kaskádní formě. Kaskádní forma obsahuje dvě přenosové funkce druhého řádu a proto se použije filtr se čtvrtým řádem (N=4).

Po určení řádu je nutné určit frekvenci pro pokles o 3dB.

$$\omega_c = \frac{2}{T} \frac{\operatorname{tg}(\omega_p T / 2)}{\sqrt[2N]{(1/A_p^2) - 1}}, \quad (5.25)$$

$$\omega_c = \frac{2}{T} \frac{\operatorname{tg}(0,04 / 2)}{\sqrt[8]{(1/0,8^2) - 1}} = \frac{1}{T} 135,21 \cdot 10^{-3}. \quad (5.26)$$

Pak se použije Butterworthova normovaná propust druhého řádu

$$H(p) = \frac{1}{p^2 + b_k p + 1}, \quad (5.27)$$

kde  $b_k$  se vypočte podle

$$b_k = 2 \sin[(2k - 1)\pi / 2N], \quad (5.27)$$

$$b_1 = 2 \sin[\pi / 8] = 0,76537,$$

$$b_2 = 2 \sin[3\pi / 8] = 1,84776. \quad (5.28)$$

Převod z normované dolní propusti na číslicovou dolní propust

$$p \rightarrow \frac{p}{\omega_c}, \quad (5.29)$$

$$H_a(p) = \frac{\omega_c^2}{p^2 + b_1 \omega_c p + \omega_c^2}. \quad (5.30)$$

Převod na diskretní číslicový filtr se provádí substitucí

$$p \rightarrow \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}}. \quad (5.31)$$

Pro jednotlivé sekce IIR filtru druhého řádu tak bude dosaženo tvaru

$$H_1(z) = \frac{\omega_c^2 T^2 (1 + z^{-1})^2}{(4 + 2b_1 \omega_c T + \omega_c^2 T^2) z^0 + (2\omega_c^2 - 8) z^{-1} + (\omega_c^2 T^2 - 2b_1 \omega_c T + 4) z^{-2}}, \quad (5.32)$$

$$H_2(z) = \frac{\omega_c^2 T^2 (1 + z^{-1})^2}{(4 + 2b_2 \omega_c T + \omega_c^2 T^2) z^0 + (2\omega_c^2 - 8) z^{-1} + (\omega_c^2 T^2 - 2b_2 \omega_c T + 4) z^{-2}}. \quad (5.33)$$

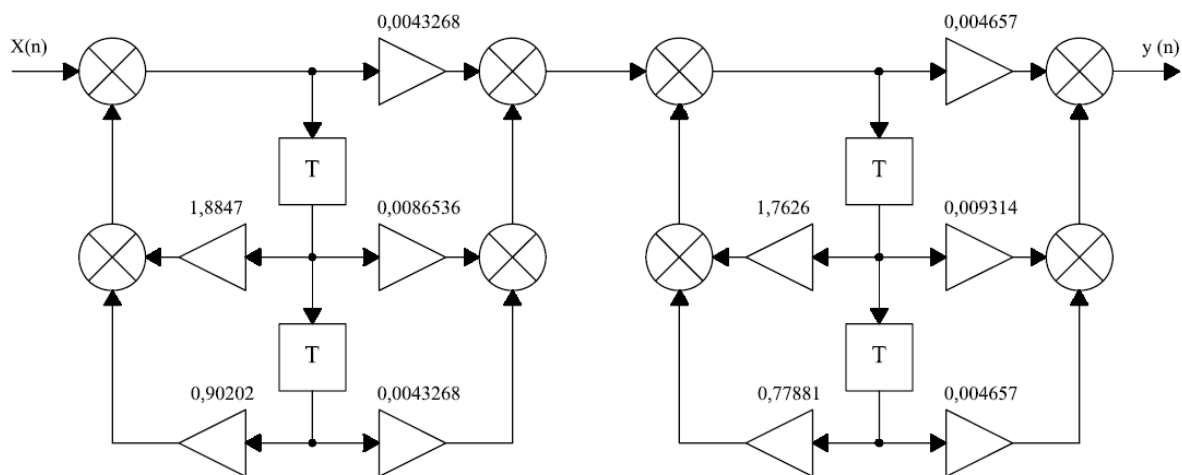
Po dosazení a úpravě rovnice lze získat přenosovou funkci filtru

$$H_a(z) = \frac{b_0 z^0 + b_1 z^{-1} + b_2 z^{-2}}{a_0 z^0 + a_1 z^{-1} + a_2 z^{-2}}, \quad (5.34)$$

$$H_1(z) = \frac{43,268 \cdot 10^{-4} (1 + z^{-1})^2}{1 + 1,8847 z^{-1} + 0,902 z^{-2}} \cdot \frac{4,225}{4,225}, \quad (5.35)$$

$$H_2(z) = \frac{46,57 \cdot 10^{-4} (1 + z^{-1})^2}{1 + 1,7626 z^{-1} + 0,7788 z^{-2}} \cdot \frac{4,517}{4,517}. \quad (5.36)$$

Získané koeficienty se pak dosadí do II. Kanonické struktury řazenou kaskádně.



**Obr. 5.5: Bloková schéma druhé kanonické struktury s koeficienty**

## 5.4 Změřené průběhy filtru IIR

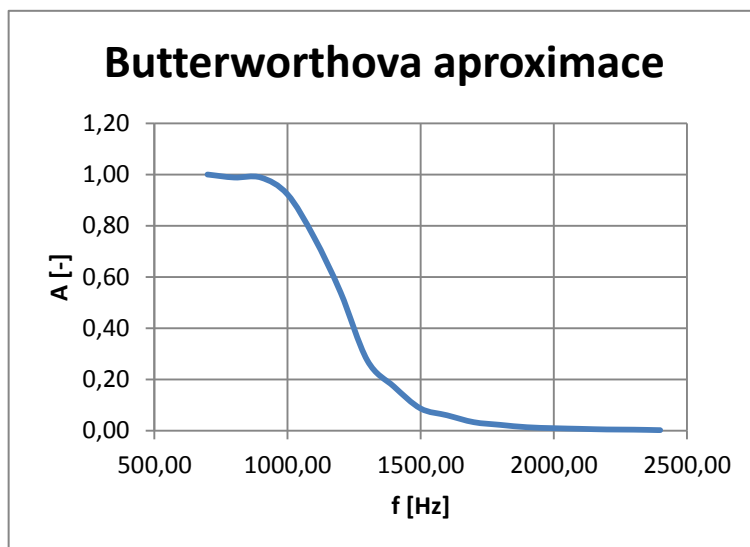
f[Hz]	U <sub>x</sub> [V]	Butterworthova aproximace	
		U <sub>y</sub> [V]	A
700,00	5,60	5,60	1,00
800,00	5,60	5,53	0,99
900,00	5,60	5,53	0,99
1000,00	5,60	5,17	0,92
1100,00	5,60	4,23	0,76
<b>1200,00</b>	<b>5,60</b>	<b>3,02</b>	<b>0,54</b>
1300,00	5,60	1,54	0,27
1400,00	5,60	0,97	0,17
1500,00	5,60	0,49	0,09
1600,00	5,60	0,34	0,06
1700,00	5,60	0,19	0,03
1800,00	5,60	0,13	0,02
1900,00	5,60	0,08	0,01
2000,00	5,60	0,06	0,01
2100,00	5,60	0,04	0,01
2200,00	5,60	0,03	0,01
2300,00	5,60	0,02	0,00
2400,00	5,60	0,01	0,00

Tab. 5.4: Naměřené hodnoty IIR filtru

Příklad výpočtu:

Zesílení:

$$A = \frac{U_y}{U_x} = \frac{5,6}{3,02} = \underline{0,54}. \quad (5.37)$$



Obr. 5.3: Frekvenční charakteristika Butterworthovy aproximace

## 6 ZÁVĚR

Předmět Signálové procesory se vyučuje už řadu let. Cílem této diplomové práce bylo popsat jeho laboratorní stanoviště a dle potřeby jej doplnit. Největším nedostatkem pro výuku předmětu jsou nedostatečně teoreticky popsané laboratorní úlohy a absence materiálu popisujícího rozšiřující desku. K aplikaci číslicových filtrů tak vyvstala povinnost popsat, jak má daný algoritmus fungovat a jaké budou jeho úskalí. Z toho důvodu se diplomová práce nejprve zaměřuje na popis procesoru TMS320F2812 používaného při výuce, kde je popsána jeho vnitřní struktura, její podoba a možnosti nastavení. Znalosti vnitřní struktury jsou pak využity k programování algoritmu pro danou laboratorní úlohu.

Diplomová práce se hlavně zaměřuje na číslicové filtry, kterým je věnována celá kapitola 3 a kapitola 4. Je zde popsáno základní rozdělení a teoretické odvození matematických vztahů. K usnadnění práce při programování filtrů byly použity knihovny filtrů, které byly vytvořeny a dodány výrobcem procesoru, firmou Texas Instrument. Knihovny jsou pro daný procesor upravené tak, že jejich optimalizace pro výpočet je na vysoké úrovni. Firma k nim poskytuje i návod na jejich implementaci. Návod je psán v anglickém jazyce. Přestože by neměl být pro studenta problém pochopit základní strukturu FIR filtru a IIR, pro potřeby výuky bylo nutné tyto texty lépe vysvětlit a doplnit o chybějící teorii.

Přílohy obsahují vytvořená zadání laboratorních úloh i s teoretickým popisem. Všechny zadané úlohy byly odměřeny a k nim vypracovány průvodní zprávy.

Cílem bylo napsat tuto diplomovou práci tak, aby student, který bude tyto texty k výuce předmětu Signálové procesory používat, je snadno pochopil a mohl tak své nabyté teoretické znalosti posléze využít v praxi. Práce pro mne byla velmi přínosnou, proto doufám, že takto získané nové znalosti využiji a následně dále rozšířím v praxi.

## 7 SEZNAM POUŽITÉ LITERATURY

- [1] Zdeněk Smékal, Petr Sysel : Signálové procesory. Vyd. Praha: Sdělovací technika, 2006  
ISBN 80-86645-08-8
- [2] Doc. Ing. Zdeněk Smékal CSc.: Číslicové filtry vyd. VUT Brno, 1993, ISBN 80-214-0500-7
- [3] , DrSc., doc. RNDr. Miroslav Vlček, DrSc.: Analogoví a Číslicové filtry. vyd. ČVUT Praha, 2000, ISBN 80-01-02178-5
- [4] Doc. Ing. Vratislav Davídek, CSc.: Implementace algoritmů číslicového zpracování signálů v reálném čase. Vyd. ČVUT Praha. 2004, ISBN 80-01-03114-4
- [5] Doc. Ing. Petr Skalický, CSc.: Aplikace signálových procesorů. vyd. ČVUT praha, 2002, ISBN 80-01-02647-7
- [6] Robert Vích, Zdeněk Smékal: Číslicové filtry. vyd. Academia Praha 2, 2000, ISBN 80-200-0761-X
- [7] Doc. Ing. Vratislav Davídek, Doc. Ing. Pavel Sovka CSc.: Číslicové zpracování signálů a implementace, vyd. ČVUT Praha, 1996, Skriptum ČVUT FEL
- [8] Doc. Ing. Zdeněk Smékal, CSc., Ing. Petr Sysel: Číslicové filtry. Skripta VUT Brno
- [9] Texas Instrument. *Data manual* [online]. 2011 [cit. 2011-2-12]. Dostupný z WWW: <<http://focus.ti.com/lit/ds/symlink/tms320f2812.pdf>>
- [10] Texas Instrument. *Filter Library* [online]. 01-12-2002 [cit. 2011-2-12]. Dostupný z WWW: <<http://focus.ti.com/docs/toolsw/folders/print/sprc082.html>>
- [11] Texas Instrument. *Module 14 : C28x FIR - Filter* [online]. 01-12-2002 [cit. 2011-2-12]. Dostupný z WWW: <[www.ti.com/ww/cn/uprogram/share/ppt/c2000/Part14.ppt](http://www.ti.com/ww/cn/uprogram/share/ppt/c2000/Part14.ppt)>
- [12] Texas Instrument. . *Datasheet DAC7715* [online]. 2009 [cit. 2011-2-12]. Dostupný z WWW: <<http://focus.ti.com/lit/ds/symlink/dac7715.pdf>>

## 8 SEZNAM PŘÍLOH

Příloha č. 1: Rozšiřující deska ze strany plošných spojů

Příloha č. 2: Rozšiřující deska ze strany součástek

Příloha č. 3: Seznam součástek

Příloha č. 4: Obvodové schéma upavující signál pro A/D převodník

Příloha č. 5: Obvodové schéma D/A převodníku

Příloha č. 6: Obvodové schéma napájecí části

Příloha č. 7: Knihovna IIR16.asm

Příloha č. 8: Knihovna FIR16.asm

Příloha č. 9: Knihovna DAC7715

Příloha č. 10: Program pro filtr IIR

Příloha č. 11: Program pro filtr FIR

Příloha č. 12: Obrázek laboratorního stanoviště

Příloha č. 13: Zadání protokolů

Příloha č. 14: Vzorový protokol k úloze č.1

Příloha č. 15: Vzorový protokol k úloze č.2

Příloha č. 16: Vzorový protokol k úloze č.3

Příloha č. 17: Vzorový protokol k úloze č.4